L Number	Hits	Search Text	DB	Time stamp
3	54	(700/220).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 08:59
4	602	(700/227,223,226,222,252).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 08:59
5	3284	(709/206,217).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:05
6	976	(53/474,157).CCLS.	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:05
7	4901	((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:06
8	2026	(((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and (mail)	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:09
10	0	((((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and (mail)) and ((((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and printstream)	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:06
9	4	(((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and printstream	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:06
12	0	((((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and printstream) and ((((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and "mail inserter")	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:10
11	8	(((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and "mail inserter"	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:24
13	2	(((((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and printstream) or ((((700/220).CCLS.) or ((700/227,223,226,222,252).CCLS.) or ((709/206,217).CCLS.) or ((53/474,157).CCLS.)) and "mail inserter")) and ("reprint" or "re-print")	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:26
14	2	("6119051").PN.	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:26
15	1	(("6119051").PN.) and reprint	USPAT; US-PGPUB; EPO; JPO; DERWENT	2003/06/16 09:26

US006119051A

United States Patent [19]

Anderson, Jr. et al.

[11] Patent Number: 6,119,051

[45] Date of Patent: Sep. 12, 2000

[54] **CLIENT-SERVER SYSTEM, METHOD AND COMPUTER PRODUCT FOR MANAGING DATABASE DRIVEN INSERTION (DDI) AND MAIL PIECE TRACKING (MPT) DATA**

[75] Inventors: **Ralph R. Anderson, Jr.**; **Mark G. Mackelprang**, both of Tucson, Ariz.

[73] Assignee: **Bell & Howell Mail and Messaging Technologies Co.**, Durham, N.C.

[21] Appl. No.: **09/183,811**

[22] Filed: **Oct. 30, 1998**

Related U.S. Application Data

[60] Provisional application No. 60/105,804, Oct. 27, 1998.

[51] **Int. Cl.**[7] **G06F 7/00**

[52] **U.S. Cl.** **700/221**; 700/226

[58] **Field of Search** 700/220, 221, 700/223, 224, 226; 395/200.33; 270/58.31, 52.19; 209/584, 583, 900, 939

[56] **References Cited**

U.S. PATENT DOCUMENTS

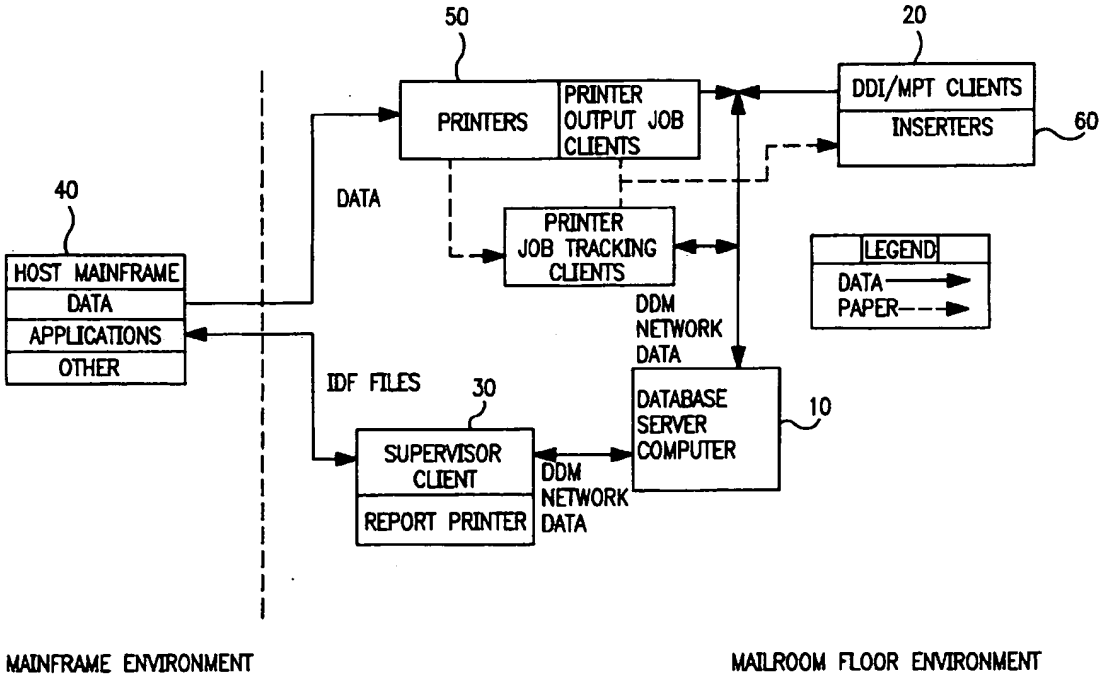
4,527,468	7/1985	Piotroski	101/2
4,800,505	1/1989	Axelrod et al.	364/478
4,817,042	3/1989	Pintsov	364/478
4,947,333	8/1990	Sansone et al.	364/464.02
4,999,481	3/1991	Baer et al.	235/375
5,419,440	5/1995	Picoult	209/583
5,469,576	11/1995	Dauerer et al.	395/186
5,612,888	3/1997	Chang et al.	364/478.09
5,777,883	7/1998	Lau et al.	364/478.08
5,873,073	2/1999	Bresnan et al.	705/410
5,918,220	6/1999	Sansone et al.	705/408

Primary Examiner—Jospeh E. Valenza
Assistant Examiner—Khoi H. Tran
Attorney, Agent, or Firm—Jenkins & Wilson, P.A.

[57] **ABSTRACT**

A client/server architecture for database driven insertion and mail piece tracking system, method, and computer program product is disclosed. A database is populated with database driven insertion data comprising instructions for handling mailpiece material. A server manages the database by responding to requests for mail processing instructions from clients and storing mailpiece data received from clients. A scanning device reads key code marked mailpiece material in which the key code corresponds to a database location containing instructions for handling mailpiece material. A client processor receives the key code from the scanning device, and transmits a request to the server for accessing the database location containing the instructions for handling mailpiece material. The server retrieves the instructions for handling mailpiece material, and transmits the instructions to the client. The client causes the performance of a mail processing task in accordance with the instructions, gathers mailpiece tracking data as the mailpiece material is processed, and forwards mailpiece tracking data to the server. The database information is accessible to report writing and generating software applications which cull data pertaining to a given mail processing job into a desired format.

20 Claims, 1 Drawing Sheet



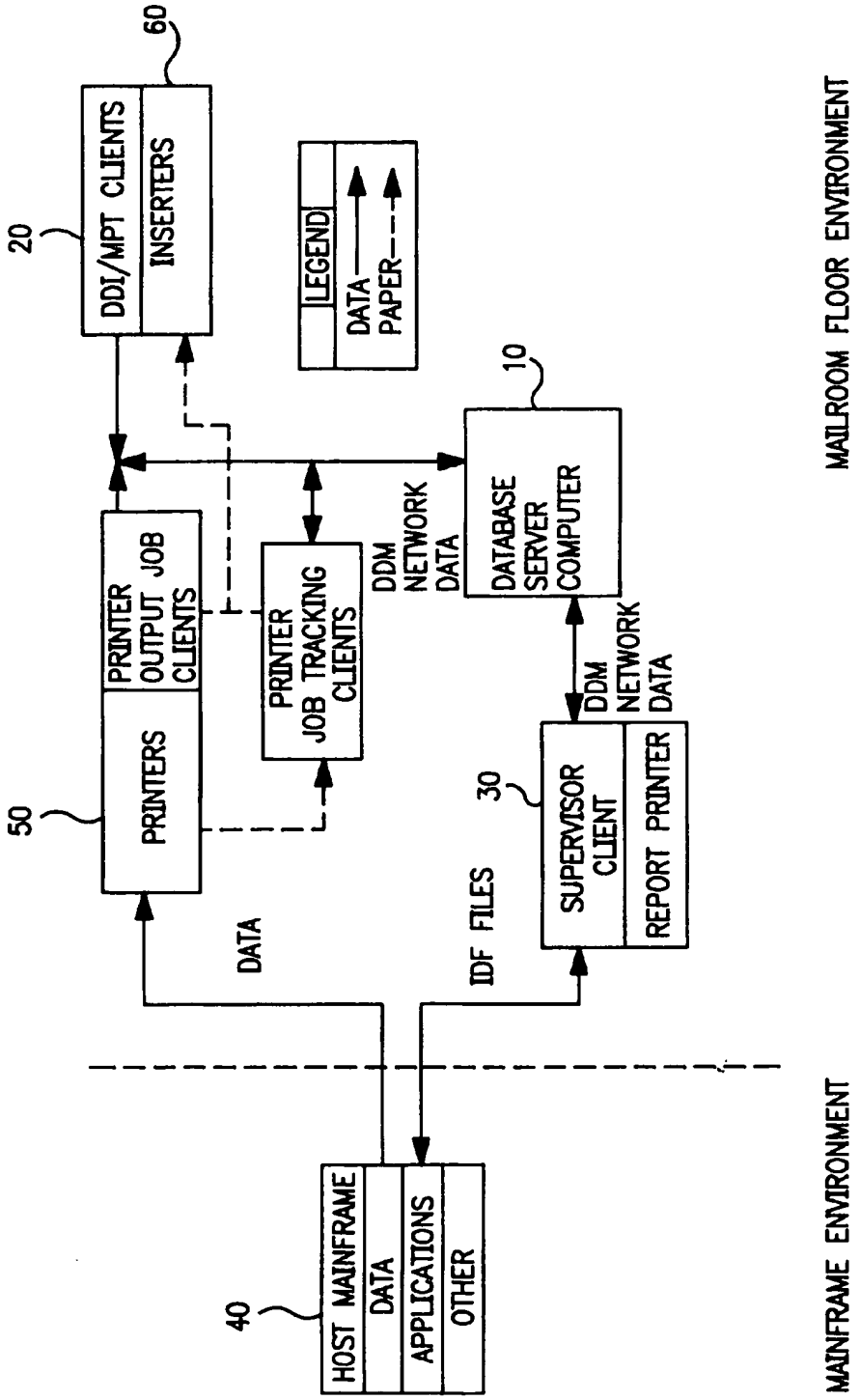

60
DDI/MPT CLIENTS
INSERTERS
20
LEGEND
DATA
PAPER
10
DATABASE SERVER COMPUTER
DDM NETWORK DATA
PRINTER OUTPUT JOB CLIENTS
PRINTER JOB TRACKING CLIENTS
PRINTERS
50
DDM NETWORK DATA
30
SUPERVISOR CLIENT
REPORT PRINTER
DATA
IDF FILES
MAILROOM FLOOR ENVIRONMENT
40
HOST MAINFRAME DATA
APPLICATIONS
OTHER
MAINFRAME ENVIRONMENT

FIG. 1

CLIENT-SERVER SYSTEM, METHOD AND COMPUTER PRODUCT FOR MANAGING DATABASE DRIVEN INSERTION (DDI) AND MAIL PIECE TRACKING (MPT) DATA

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to and claims the benefit of the U.S. Provisional Patent Application entitled "A Client-Server System and Method Of Managing Database Driven Insertion (DDI) and Mail Piece Tracking (MPT) Data", filed on Oct. 27, 1998, 60/105,804.

FIELD OF THE INVENTION

The present invention relates generally to manufacturing environments that wish to relate large amounts of information to a small identifier. More specifically, the present invention relates to a client-server system, method, and computer program for managing database driven insertion (DDI) and mail piece tracking (MPT) data for holding and managing mailroom data in a consistent and easy to use manner.

BACKGROUND OF THE INVENTION

Currently, it is common in mail processing for mail piece data to be handled utilizing a file-based system (i.e. using a flat ASCII file to hold all database driven insertion and mail piece tracking information). A client/server concept involves replacing flat files with a database server which maintains indices and relations between various data fields, as described further hereinbelow. Also as described further hereinbelow, utilizing a client-server concept, as according to the present invention, allows an interface to be developed for client programs to be able to read database driven insertion (DDI) data from the database and write mail piece tracking data back to the database.

Database driven insertion (DDI) is currently being accomplished in conventional mail processing by storing mail processing instructions in a flat ASCII file, reading an account number from paper via a laser scanner, calculating the offset of the data in the file that corresponded to the account number read, and reading the data at that offset point into the mail processing equipment. Mail piece tracking has been accomplished by storing information about a mailpiece back into the database driven insertion (DDI) file, or possibly a separate file whenever the mailpiece processing was complete. This was, and still is, the industry norm because it is believed that a database is not capable of keeping up with the read and write rates required for multiple mail processing machines. In contrast to this norm, the present invention, however, can and does keep up with the read and write rates required for multiple mail processing machines using the aforementioned client/server concept, as described further hereinbelow.

Database driven insertion (DDI) data typically describes to individual mail processing inserters which inserts to feed, how many sheets are in an account, what actions the inserter is to perform on the account, what address should be printed on the envelope, and/or other information as apparent to those of skill in the art.

Mail piece tracking (MPT) data typically describes what actually happened to the account during processing, i.e. what machine processed it, when the machine started processing it, when the machine finished processing it, which operators were running the machine, which inserts fed, and/or other information as apparent to those of skill in the art.

Using a database under a client/server architecture (as opposed to a flat ASCII file) for insertion and tracking has many significant advantages which will be readily appreciated by those of skill in the art. Clients (which can comprise mail inserters, mail sorters, printers, other applications, and/or other suitable clients as recognized by those of skill in the art of mail processing) can request and receive only the information they need which decreases the overall load borne by the communications network. Other clients (report generators) can create reports much easier with well known database reporting tools. The server provides a common repository for all mail piece tracking and database driven insertion data, which, in turn, allows management from one computer and location, i.e. centralized operation. The database server provides excellent file locking and read/write contention protection superior to that of ASCII flat files. The server also provides services to inform clients whether a record was updated "underneath" it. This provides site-wide duplicate checking for all mailpieces to ensure there are no duplicate mailpieces being processed. Additionally, the database server enforces data consistency. The server will not allow clients to write "invalid" data into the database. This is very difficult to enforce in file-based systems. The server further provides "stored procedures" which allow the server to change its functionality without necessarily modifying client code. Other advantages can also exist as recognized by those skilled in the art.

In view of the above, there remains much room for improvement in the art, particularly for a new system and method of "publishing" and "recording" database driven insertion and mail piece tracking data.

DISCLOSURE OF THE INVENTION

In accordance with the present invention, a novel client-server system, method, and computer program for managing database driven insertion (DDI) and mail piece tracking (MPT) data for holding and managing mailroom data in a consistent and easy to use manner is provided. "Managing" of data according to the present invention refers to a system that controls, utilizes, tracks, and reports on all aspects of database driven insertion and mail piece tracking data. By the client/server database architecture for managing database driven insertion and mailpiece tracking in a mail processing environment according this invention, a customer initially sets up a mail processing site by defining within the client/server architecture running database driven insertion and mail piece tracking system parameters such as Users, Privileges, JobSetups, Materials, etc., before any actual mail processing occurs. Next, the customer generates data (generally in a mainframe environment) that is intended to be printed and mailed. The data is run through a utility like Bell & Howell's *Transformer*™ or their own custom software to create a "side file" that contains the database driven insertion information required by a mail processing insertion device. Each print run has a matching side file generated for it. Material is printed and the side file is loaded/inducted into the database driven insertion and mail piece tracking system. The customer physically conveys the printed material to the inserter, loads the mail processing job currently programmed, places the materials called for by the mail processing job (e.g., inserts, printed materials, envelopes, etc . . .) into the correct locations, and begins running the mail processing job. As a mail processing inserter reads each reader code or key that has been strategically placed on the mailpiece materials, the inserter makes a request for the database driven insertion data associated with that particular key from the database. The database

sends the insertion data back to the inserter, which uses the data to determine what actions to perform on this particular account. As each mailpiece leaves the inserter, mail piece tracking data is written into the database associated with each database driven insertion record that records, for instance, the Machine, Operators, Time, Date, JobSetup, Inserts Fed, etc., for each mailpiece.

It is therefore an object of the present invention to provide a novel client-server system, method, and computer program for managing database driven insertion (DDI) and mail piece tracking (MPT) data for holding and managing mailroom data in a consistent and easy to use manner.

It is another object of the present invention to store all types of data in the database driven insertion server that are related to the other types of data in a way that makes generating very flexible and detailed reports very easy.

It is a further object of the present invention to be able to modify instructions regarding the processing of each mailpiece right up until the time the mailpiece is placed on a machine for processing.

It is a still further object of the present invention to generate a standard postal manifest that details all pieces processed and the amount owed the post office.

It is a still further object of the present invention to re-produce a list of mailpieces processed properly and mailpieces that did not process properly.

Some of the objects of the invention having been stated, other objects will become evident as the description proceeds, when taken in connection with the accompanying drawings described below.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing advantages and features of the present invention will be appreciated more fully from the following description with reference to the accompanying drawings in which:

FIG. 1 illustrates a client/server architecture capable for use with the present invention.

BEST MODE FOR CARRYING OUT THE INVENTION

The present invention now is described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

Referring now to FIG. **1**, one possible client/server architecture is shown which includes a database server computer **10** used as the central repository of all data, a machine client computer (console) **20**, a supervisory computer (supervisor) **30**, and a computer network for operatively linking everything together. Solid lines represent electronic data flow while dashed lines represent physical paper or material flow throughout FIG. 1. The preferred embodiment presently uses Microsoft Windows™ NT Server 4.0 software, Interbase™ Server 5.0, and custom written software running on the server machine and Interbase™ client software and/or custom written software running on the client machines. The hardware is generally Intel Pentium™ II class generic personal computer boxes.

It is to be understood that the present invention illustrated herein is readily implementable by those of ordinary skill in the art as a computer program product having a medium with a computer program embodied thereon. The computer program product is capable of being loaded and executed on the appropriate computer processing device(s) in order to carry out the method or process steps described.

Still referring to FIG. **1**, applications on the mainframe side send print images from a host mainframe **40**, for instance, to printers **50**, and IntellaSert™ Data File (IDF) data to the database server computer **10**. Once the material is printed on by the printers **50** (which can be monitored by a reconciliation station), the printed paper is presented to mail processing finishing equipment, such as, for instance, mail processing inserters **60**. The mail processing finishing equipment **60** requests information about the accounts it is about to process from the database server **10**, using a small key encoded in the account barcode, and uses the information in the data file to continue processing the account. When the account has been completely processed (either rejected, removed, or ready to mail), the finishing equipment **60** updates the database with a complete disposition of the account. The exact status and location of each account is available at all times to users having access to the supervisor client computer **30**. Once processing has been completed, the supervisor client computer **30** can create a manifest to present to the United States Postal Service (USPS), and for any pieces that were destroyed during processing, it can feed the pertinent data back to the host to generate reprint material and new IDF data. Alternately, supervisor client computer **30** can send data to a local "Winserter"-type mail processing device to create reprints locally. This allows accounts to be handled in a totally "closed loop" fashion.

The description of the present invention describes services provided by the database server computer **10** and application interfaces provided for client applications. These services are intended to provide all the basic services available in the software system design, including data file, database driven insertion, historical reports, real time monitoring of machinery, operators, jobs, shifts, inserts tracking and chargeback, manifesting, reprinting, and/or other suitable services apparent to those of skill in the art, while adding the ability to significantly extend the feature set, all without harming backwards compatibility.

A dataset, according to the present invention, is a named compilation of related data stored on the server. Datasets are composed of ordered records, which are accessed by a record identifier. Conceptually, datasets can be envisioned as virtual files which support normal file services such as create file, open file, close file, delete file, read record, write record, and append record. Additionally, datasets have the ability to delete records, provide multiple views of records, create a new dataset based on an existing dataset, and some search criteria among other abilities. All datasets have one thing in common, namely, each dataset record has an attribute called "RecordID". The "RecordID" field defines the order of records in a dataset. The attribute "RecordID" may be stored inside the record, or may be implicitly designed by the dataset itself. In either case, users of a dataset need only know that every record "knows" its position, and every dataset "knows" its order.

A record is the basic element of a dataset. This is the smallest element that can be modified in a dataset. Note that a record from a client point of view, and a record from a server point of view may be different for both the read and write cases. Clients may view a record as only a very small number of fields, whereas the server may actually have many fields for every record. As long as the client fields are a subset of the server fields, the server will send only the fields requested back to the client.

A RecID is the basic "key" column for any dataset. The word "key" is emphasized, because this in no way implies that datasets are indexed databases. It is meant to infer the function of a key field. All dataset records have a RecordID which starts at 1, and increases sequentially allowing elements of the dataset to be accessed by clients using the read record, update record, delete record, insert record, append record, open dataset, close dataset, seek record, and tell record type methods available in the standard "C" File/IO function set. Note that the actual order of data records in the dataset is both unknown and irrelevant. Unknown because the server can implement it in any way it chooses, and irrelevant because the server's only constraint on returning the dataset record to the client is that it happens "fast enough".

Views are defined by the services layer to provide data of interest from a dataset. A view defines all the fields needed from a record in a dataset. A record in a dataset can have many views defined simultaneously, and the data needed by the client defines which view is used. There are two (2) main uses for views in the client services. In the case of reading records from a dataset, the view defines the set of fields the client wants the server to return for each record read. In the case of writing records from a dataset, the view defines the set of fields the client must send to the server for each record written.

DDM stands for device and data management and refers to a (set of) client and server computer(s) that contain a large set of data relating current documents and past documents, along with tools to allow management of this data. The database server computer will never serve file or print services, as its only purpose is to provide data services through a suite of applications. These applications will be network communication based.

One feature of the present invention is termed the client developers kit (CDK). It is an application programming interface which allows a client to be developed using any platform that has an *Interbase*™ client library available. The client developers kit application programming interface gives access to data of interest without having to know about or understand the details of the database.

Mail piece tracking refers to, inter alia, a client's ability to report the disposition of a mailpiece without necessarily being able to use the database driven insertion data defined in a record. This feature can be used for reprint generation and for generating manifests.

Database driven insertion and processing data file (process directive file) are terms referring generally to the concept of having a electro-mechanical piece of equipment (an inserter, for example) associate large amounts of data with a small "key" or identifier printed on the material via codes (or other machine readable method). The data referred to by the "key" is changeable up to the moment the data is read and "placed" on the equipment. The data can supply (but is not limited to) address information for printing on envelopes, which inserts to drop on this individual account, whether this account should be stapled, etc. Of particular interest is a small piece of the data that allows inserts to be targeted to accounts individually.

The term "stream" relates to input devices, such as continuous forms cutters and cut sheet feeders on a mail processing inserter. For instance, a mail processing inserter with two cutters and one sheet feeder is deemed to have three (3) streams. Hence, streamSheet**01**, streamSheet**02**, and streamSheet**03** in the data file fields are filled. By convention, the most "upstream" mail processing device is said to be stream **1**.

Another feature of the present invention is its ability to provide for duplicate checking. As the client inserter "finishes" each mailpiece, the disposition of the mailpiece is saved in the data file data set via the data file account ID. The database driven insertion client can now provide real-time duplicate checking for the client inserter. If any other machine on the network has processed or is currently processing the mailpiece in question, the "latest" copy of the mailpiece will be deemed duplicate. A warning message will print on the client computer screen, and the mailpiece will be targeted for the reject bin.

It always has been and will always be possible for a printer operator or other worker(s) on the mailroom floor to introduce duplicate copies of already existing material into the processing environment. To detect and remedy these problems as soon as possible, the data file (IDF) system includes real-time duplicate checking software. Overall, there should be no instances where the data file system does not detect a duplicate account. In nearly all cases, it will detect and reject them in real-time. In some cases where duplicate accounts are being processed within one (1) minute of each other on different inserters within the same network, the system will not be able to warn the operator of the duplicate until the second of the duplicate accounts exits the machine.

When two or more machines process the same data with overlapping material the printer operator backs up the print job between stacks of paper. Database driven insertion clients would not be able to detect these errors by themselves, since the account sequencing information would be correct. Depending on how close in time the various mail processing machines processed the material, this case would be caught either by the "Server Reads Data" case or the "Server Writes Data" case.

Should a stack of material on a single machine have duplicate material (from a printer rollback, for example) in the middle of the stack, the database driven insertion client would catch the first duplicate, because the account sequence there would be invalid. If more than one account were duplicated, however, the rest of the accounts would process normally. Duplicate checking detects this problem in the "Client Receives Data" case, the "Server Reads Data" case, or the "Server Writes Data" case, depending on the timing.

In the "Server Reads Data" case, when the server receives a request for an account record, it checks the final destination field of that record. If it is 'NP' (not processed), 'OR' (operator removed), or 'R2' (reject bin), the server changes nothing and passes the data record down to the client for processing. If the final destination is anything different than those mentioned above, the server sets the target destination of the account to 'DP' (duplicate), which will result in the account being sent to the reject bin. The client, whenever it receives a 'DP' target destination, can inform the operator that a duplicate account will be rejected.

In the "Server Writes Data" case, when the server receives data back from the client to write into the data file database, it will know whether the record in the database has been modified. If it has been modified, the server checks to see if the final destination is set to an invalid destination. If it is, it will set the final destination of the record to 'DP' (duplicate), and send a message to the client to inform the operator that a duplicate mailpiece exists.

In the "Client Receives Data" case, when the client receives a record from the server, it checks all the accounts that it is currently processing. If it finds a matching account,

it will set the target destination of the new duplicate account to 'DP'. This account will eventually go to the reject bin.

The abbreviations used in the tables below are explained defined as:

SH	Standard Handling (The destination(s) for "Good" mailable mail).
SD	Security Divert. (The destination(s) for "Special" mail)
OW	Overweight Divert. (The destination(s) for material that is too heavy or too thick to be mailed).
RX	Reject Divert. (The destination(s) where "bad" or damaged material is sent).
OR	Operator Removed. (The destination where material that is removed by the operator is sent).
NP	The initial or Not Processed destination. This flag indicates the mailpiece must be recreated.
DP	Duplicate Account. This indicates that the account was processed at least twice (i.e. more than one copy of this account went to 'SH', 'SD', or 'OW'.
LH	Late Hold. This indicates that the user (via a pre-processing function) has determined that the account should not be processed, and wants to require the inserter operator to remove the account from the mailing.

When the client reports a finished account to the server, the server determines the final disposition of the mailpiece by comparing the "current" disposition with the "new" disposition. Based on these two values, it chooses to increment (or not) a value called the "Duplicate Count" (this is the first value in each cell in the table below) and decides whether to save the "new" data into the table (the second value in each cell of the table below). Lastly, the server returns a status for every write, and if the status is affected by the destinations, the status is listed in the third row of each cell. The following table of new and existing final destinations describes the rules governing every possible new and existing final destination:

TABLE 1

Duplicate Destinations

	EXISTING FINAL DESTINATION						
	SH	SD	OW	RX	OR	NP	LH
SH	1 No ERR_DUP	1 No ERR_DUP	1 No ERR_DUP	0 Yes ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	1 No ERR_LH
SD	1 No ERR_DUP	1 No ERR_DUP	1 No ERR_DUP	0 Yes ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	1 No ERR_LH
OW	1 No ERR_DUP	1 No ERR_DUP	1 No ERR_DUP	0 Yes ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	1 No ERR_LH
RX	0 No ERR_NON	0 No ERR_NON	0 No ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	0 No ERR_NON
OR	0 No ERR_NON	0 No ERR_NON	0 No ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	0 Yes ERR_NON	0 No ERR_NON

When data file data is read from the database, if the duplicate count of the record is greater than zero, the final destination is returned as 'DP', regardless of what the actual final destination in the data is. The only exception to this is where the final destination is 'LH'. In this case, the final destination returned is 'LH', regardless of what the actual duplicate count is. The following table delineates these rules:

TABLE 2

Duplicate Destination Read Rules

	FINAL DESTINATION						
	SH	SD	OW	RX	OR	LH	NONE
0	SH	SD	OW	RX	OR	LH	NP
>0	DP	DP	DP*	DP*	DP*	LH	DP*

Note that there should never be final destinations OW, RX, OR, or NONE with a duplicate count greater than zero. These cases are handled as data integrity errors.

When a user "fixes" the problem with a duplicate (or Late Hold), the client can call the "Release Duplicate" application programming interface which will decrement the duplicate count, return the current duplicate count and a status code. The table describing these rules is as follows:

TABLE 3

Release Duplicate Actions

DUPE	FINAL DESTINATION						
COUNT	SH	SD	OW	RX	OR	LH	NONE
<2	ERR_NON DC = 0	ERR_NON DC = 0	ERR_NON DC = 0	ERR_NON DC = 0	ERR_NON DC = 0	ERR_NON DC = 0	ERR_NON DC = 0
>1	ERR_DUP DC_	ERR_DUP DC_	ERR_DUP DC_	ERR_NON DC = 0	ERR_NON DC = 0	ERR_LH DC_	ERR_NON DC = 0

Note that the first item in each cell is the error code. The second is the action to be performed on the Duplicate Count (DC).

The system and methodology of the present invention can be illustrated by way of the following example, which is described for illustrative purposes only and is not intended to be exhaustive of the potential applicability of the present invention.

ILLUSTRATIVE EXAMPLE

Consider an organization that wishes to print and mail a large batch of material to a set of its customers. First, the organization generates print images within a mainframe host computer, for instance. The print images, representing all or part of the mailpiece to be sent, are forwarded to a printer or printers to be printed on documents such as paper sheet articles. Thus, the content to be mailed is converted from electronic image to physical paper ready to be manipulated in a mail processing environment. The mainframe host computer, in this example, also generates database driven insertion data that is forwarded to the organization's mail-room database server. The database driven insertion data is then inducted or imported into the database driven insertion and mail piece tracking system.

After the material has been printed and the data has been populated into the database, the mail processing machines begin processing the printed material. An operator of the mail processing machine initiates the following process:

(1) Selecting and loading a "Job" for the machine. The job is defined in the database and was created previously by a user with authority and privilege to do so. The job defines (i) reader codes printed on the material, (ii) the "mode" of the machine, (iii) which inserts are loaded into the mailing machine, and (iv) the methods of stapling, folding, printing, etc. for the machine.

(2) Physically loading the material on the mail processing machine.

(3) If the "Name" of the database driven insertion (DDI) data is not specified on the reader codes, the user must select which set of database driven insertion data to use from the database.

(4) At this point, the machine begins processing the paper, following the "Job Level" instructions contained in the Job Setup, and the "Account Level" instructions contained in the database driven insertion data.

Database driven insertion data for the following eight (8) accounts is generated by host computers and sent to the database server computer. The database server computer stores the data in the following manner:

TABLE 4

Database driven insertion Account Data

Tray ID	IDF ID	Doc ID	Target Dest	Tray Dest	DPBC	Pull Key	User Field	Proc. Dir	Str 0	Str 1	Str 2	Str 3
4464	160 "AA"	3643	"SH"	" "	"1111111 1111"	"0000000056721475"	"00000000567 21475"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3644	"SH"	" "	"111111 11111"	"0000000059049304"	"00000000590 49304"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3645	"SH"	" "	"111111 11111"	"0000000059038117"	"00000000590 38117"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3646	"SH"	" "	"111111 11111"	"0000000059052456"	"00000000590 52456"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3647	"SH"	" "	"111111 11111"	"0000000059691501"	"00000000596 91501"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3648	"SH"	" "	"111111 11111"	"0000000057681793"	"00000000576 81793"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3649	"SH"	" "	"111111 11111"	"0000000059307249"	"00000000593 07249"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0
4464	160 "AA"	3650	"SH"	" "	"111111 11111"	"0000000058294141"	"00000000582 94141"	"NNNNNNNNNYNNNNNN NNNN"	3	0	0	0

The above table data is defined as follows:

Tray ID	Information about the mailing tray the mailpiece belongs to.
IDF ID	The IDF data group this mailpiece belongs to. Generally, an IDF corresponds to a print run.
Target Dest	The desired "destination" of the mailpiece on the mailing machine. This would correspond to "SH" (Standard Handling), "SD" (Security Divert), "OW" (Overweight).
Tray Dest	Information necessary to print a tray tag.

-continued

DPBC	(Delivery Point Bar Code.) Information necessary to print the Postnet Barcode on the mailpiece.
Pull key	Customer Defined key to look up a particular mailpiece.
User Field	Customer Defined key for customer use.
Proc Dir	Processing Directives give instructions to the machine regarding whether to Staple, Seal, Drop Inserts, etc on this particular mailpiece.
Str O-Str 3	Page count information for up to three (3) streams of material. Note that these mailpieces only have pages from stream O.

Print and Verify String Data for these mailpieces appears as follows:

Print String Data

Insert Verify String Data

As the processing of the material progresses, the machine begins to send mailpiece tracking data back to the database. The data sent back for the accounts listed above could, for example, appear as follows:

TABLE 5

Returned Mailpiece Tracking Account Data

Fin. Dest	Start Time	Finish Time	Shift ID	Job ID	Weight	Postage	Key Line	Status	Dest Rsn	Inserts Fed	Seq Num	Doc ID	IDF ID	Dup Count
"SH"	10/21/1998 17:49:47	10/21/1998 17:51:07	3	821	251	" "	"	0	1	"000"	"3"	3643	160	0
"SH"	10/21/1998 17:49:47	10/21/1998 17:51:07	3	821	251	" "	"	0	1	"000"	"4"	3644	160	0
"SH"	10/21/1998 17:49:47	10/21/1998 17:51:07	3	821	251	" "	"	0	1	"000"	"5"	3645	160	0
"SH"	10/21/1998 17:49:48	10/21/1998 17:51:07	3	821	251	" "	"	0	1	"000"	"6"	3646	160	0
"OR"	10/21/1998 17:49:47	10/21/1998 17:51:07	3	821	251	" "	"	0	546	"000"	"0"	3648	160	0
"R2"	10/21/1998 17:51:16	10/21/1998 17:52:36	3	821	251	" "	"	0	546	"000"	"0"	3649	160	0
"SH"	10/21/1998 17:51:16	10/21/1998 17:52:36	3	821	251	" "	"	0	1	"000"	"7"	3650	160	0

The data for table 5 is defined as follows:

Final Destination	The location the mailpiece ended up in on the machine.
Stad Time	The time the mailpiece began processing on the machine.
Stop Time	The time the mailpiece exited the machine.
Shift ID	The shift the mailpiece was processed on.
Job ID	The Job Instance the mailpiece was processed on.
Weight	The final weight of the mailpiece.
Postage	The final cost of the mailpiece.
Keyline	The keyline printed on the mailpiece (if any).
Status	The final status of the mailpiece.
Destination Reason it did.	The "reason" the mailpiece went to the destination
Inseds Fed	Information about which inserts fed on the mailpiece, and explanations of why.
Sequence Number	The sequence number of the mailpiece.
Document ID	Used to look up/relate DDI data in the previous table.
IDF ID	Used to look up/relate DDI data in the previous table.
Duplicate Count	Used to check for, and signal duplicate accounts.

Table 5 shows that mailpieces **3643**, **3644**, **3644**, **3645**, **3646**, and **3650** went to destination SH (the "normal" mailable destination), mailpiece **3647** was never "seen" by the machine (because of a read error, for example), **3648** was OR (operator removed) for reason #546 (possibly a jam or some other problem), **3649** was diverted to the R2 (reject bin) for the same reason (#546). Table 5 also shows that the mailpieces were processed during Shift **3** and JobInstance **821**. The database contains detailed information about the processing in the Job and Shift tables.

Once the machine finishes processing the mailpieces, reports are generated that show which mailpieces were successful, which need to be reprinted, etc. The reports are fed back into the system to start another print run.

The present invention provides several advantages over prior art systems and methods. First, all types of data stored in the database driven insertion server are related to the other types of data in a way that makes generating very flexible and detailed reports very easy.

Second, since instructions about each mailpiece are stored in the database, the instructions can be modified right up until the time the mailpiece is placed on a machine for processing. This is sometimes referred to as late binding.

Third, since all mail piece tracking data is kept in the database, one of the reports that can be generated is a standard postal manifest that details all pieces processed and the amount owed the post office. This is sometimes referred to as machine based manifesting.

Fourth, since the mail piece tracking data tracks all mailpieces processed properly and all mailpieces processed improperly, a list of mailpieces to re-produce is easy to produce. This is sometimes referred to as reprint generation.

Fifth, the database contains a physical description (including a scanned image) of all materials to be used in the mailroom. This includes inserts, envelopes, and sheets (of paper). No other mail processing implementation known to the inventors has the ability to show an image of the insert/envelope selected. This feature reduces operator errors by showing the operators pictures of the materials they should be loading into the machine. This is sometimes referred to as centralized materials data.

Sixth, the database contains information about all the machines connected to it and the instructions to the machines for each job. Thus, there is no need to program each machine separately. This is sometimes referred to as centralized job programming.

Seventh, the database contains a list of all defined "barcodes". When the user programs a job, he/she has the option of creating a new "barcode" map, or selecting one of the

already defined ones. There is no need to program the reader map on each individual machine. This is sometimes referred to as centralized reader code map programming.

Eighth, since all mail piece tracking data is in the same database, production reports can be easily generated to show relationships between different machines, operators, shifts, and jobs. This is sometimes referred to as centralized production/efficiency reports.

Ninth, since the mail piece tracking data tells which inserts all fed for each account, and contains the physical descriptions of the inserts, a report detailing the chargeback amounts can be produced. This is sometimes referred to as centralized inserts chargeback reports.

Tenth, descriptions of each user and each user's allowed privileges is kept in the database, and is managed from a single application. This allows management of all operators/users in the mailroom from one central location. This feature allows some (well trained) users to have privileges to perform In certain actions with the equipment that other (less well trained) operators would not. The allowed privileges for each user/operator is managed completely by the customer. This is sometimes referred to as centralized user privilege management.

Eleventh, descriptions of each machine are kept in the database. This allows programs like Job Setup to ask questions pertinent only to the machines the job is intended for. It also allows easy access to information about each machine without having to look at the machine computer itself. This is sometimes referred to as centralized machine definition.

Twelfth, the database contains a master event log that contains all events that may be of interest to a user/customer. These events include (but are not limited to) Machine Starting, Machine Stopping, User Logged In, User Logged Out, Job Started, Job Ended, Shift Started, Shift Ended, Job Created, Job Deleted, Job Modified, etc. This is sometimes referred to as a centralized event log.

Appropriate computer program code in combination with hardware implements many of the elements of the present invention. This computer code is often stored on storage media. This media can be a diskette, hard disk, CD-ROM, or tape. The media can also be a memory storage device or collection of memory storage devices such as read-only memory (ROM) or random access memory (RAM). Additionally, the computer program code can be transferred to the appropriate hardware over some type of data network.

The foregoing is illustrative of the present invention and is not to be construed as limiting thereof. Although a few exemplary embodiments of this invention have been described, those skilled in the art will readily appreciate that many modifications are possible in the exemplary embodiments without materially departing from the novel teachings and advantages of this invention. Accordingly, all such modifications are intended to be included within the scope of this invention as defined in the claims. For instance, the architecture described herein is easily extendible to manage processes not normally associated with the mailroom. Some of these processes include direct billing over the internet, print on demand, archiving collections of documents to a CD-ROM, etc.

In the claims, any means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures. Therefore, it is to be understood that the foregoing is illustrative of the present invention and is not to be construed as limited to the specific embodiments disclosed, and that modifications to the disclosed embodiments, as well as other embodiments, are intended to be included within the scope of the appended claims. The invention is defined by the following claims, with equivalents of the claims to be included therein.

That which is claimed:

1. A system for managing abase mailpiece processing comprising:

(a) a server computer including a database comprising a plurality of records including instruction sets for handling individual mailpieces;

(b) at least one reader for reading key code marked mailpieces each key code corresponding to one instruction set of the instruction sets in the database; and

(c) a client computer, responsive to the reader, for requesting the instruction set for handling a mailpiece corresponding to a key code on the mailpiece as the key code is read and causing performance of at least one mail processing task in accordance with the instruction set, the client computer immediately updating the record in the server computer database corresponding to the mailpiece being processed as the at least one mail processing task is performed to indicate the status of the mailpiece and track the mailpiece in real time.

2. The system of claim 1 comprising a supervisor computer coupled to the server computer for generating at least one report concerning the performance of at least one mail processing task.

3. The system of claim 2 wherein the supervisor computer requests mailpiece tracking data from the server computer and generates at least one report concerning the tracking of at least one mailpiece.

4. The system of claim 3 in which the at least one report is a postal manifest report.

5. The system of claim 1 wherein the database comprises stored instructions about each mailpiece and wherein the instructions are modifiable at any time prior to performance of a mail processing task utilizing the instructions.

6. A system for managing database driven insertion and mailpiece tracking data comprising:

(a) a server populating a database with data comprising a plurality of records including instruction sets for handling individual mailpieces;

a reader for reading, from a mailpiece, a key code corresponding to a database location containing an instruction set for handling the mailpiece;

(c) a client, responsive to the reader, for requesting the instruction set for handling the mailpiece from the server, such that the server receives the request from the client, accesses the instruction set for handling the mailpiece and forwards the instruction set to the client;

(d) a mail processing device coupled to the client for performing at least one mail processing task on the mailpiece in accordance with the instruction set; and

(e) means, responsive to the mail processing device, for immediately updating a record in the database corresponding to the mailpiece being processed.

7. The system of claim 6 further comprising means for generating at least one report concerning the performance of at least one mail processing task.

8. The system of claim 6 further comprising means for generating at least one report concerning the tracking of at least one mailpiece.

9. The system of claim 8 in which the at least one report is a postal manifest report.

10. The system of claim 6 in which said database may be populated with new data anytime prior to performance of a mail processing task utilizing said data.

11. A method for managing database driven insertion and mailpiece tracking data comprising:

(a) populating a database with data comprising a plurality of records including instruction sets for handling individual mailpieces;

(b) reading, from a mailpiece, a key code corresponding to an instruction set for handling the mailpiece;

requesting the instruction set for handling the mailpiece from the database;

(d) at a server:

(i) receiving requests from one or more clients for instruction sets for handling individual mailpieces and, in response, accessing the requested instruction set for handling the mailpiece from the database; and

(ii) forwarding the requested instruction set to the one or more clients; and

(e) at a client:

(i) receiving the requested instruction set for handling the mailpiece from the server;

(ii) performing at least one mail processing task in accordance with the requested instruction set;

(iii) gathering mailpiece tracking data as the mailpiece is processed during said at least one mail processing task; and

(iv) immediately updating a record in the database corresponding to the mailpiece being processed.

12. The method of claim 11 further comprising generating at least one report concerning the performance of at least one mail processing task.

13. The method of claim 11 further comprising generating at least one report concerning the tracking of at least one mailpiece.

14. The method of claim 13 wherein the at least one report is a postal manifest report.

15. The method of claim 11 comprising modifying the instructions in the database immediately before accessing the database to retrieve the instructions.

16. A client/server system for managing mail processing and mailpiece tracking data, the system comprising:

(a) a database server computer comprising a central repository for mail processing data including instruction sets for processing individual mailpieces and mailpiece tracking data for tracking the individual mail pieces;

(b) a machine client computer for requesting one of the instruction sets for processing an individual mailpiece from the database server computer based on account information read from a mailpiece and for transmitting updates to the database server computer in real-time as the mailpiece is processed;

(c) a supervisory computer for communicating with the database server computer and the machine client computer to allow users to determine status and location information relating to mailpieces being processed; and

(d) a network for linking the database server computer, the machine client computer, and the supervisory computer.

17. The client/server system of claim 16 wherein after processing of the account has been completed, the supervisory computer generates a manifest indicative of the processing of the account.

18. The client/server system of claim 16 comprising a mainframe computer coupled to the network for sending mail processing data to the database server computer.

19. The system of claim 16 wherein the database server computer is adapted to display a scanned image of the mailpiece being processed.

20. The system of claim 16 wherein the database server computer is adapted to allow late binding of the account information from the mailpiece to one of the instruction sets.

* * * * *

US005928323A

United States Patent [19]

Gosling et al.

[11] Patent Number: 5,928,323

[45] Date of Patent: Jul. 27, 1999

[54] **APPARATUS AND METHOD FOR DYNAMICALLY GENERATING INFORMATION WITH SERVER-SIDE SOFTWARE OBJECTS**

[75] Inventors: **James A. Gosling**, Woodside; **Pavani Diwanji**, San Jose; **David W. Connelly**, Los Altos, all of Calif.

[73] Assignee: **Sun Microsystems, Inc.**, Mountain View, Calif.

[21] Appl. No.: **08/828,318**

[22] Filed: **Mar. 28, 1997**

Related U.S. Application Data

[63] Continuation-in-part of application No. 08/756,325, Nov. 25, 1996, abandoned, which is a continuation of application No. 08/657,712, May 30, 1996, abandoned.

[51] **Int. Cl.**[6] **G06F 13/00**

[52] **U.S. Cl.** **709/203**; 709/219; 709/217

[58] **Field of Search** 395/200.49, 200.33, 395/200.47, 200.46, 200.56, 200.55, 200.59, 200.42

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,341,477	8/1994	Pitkin et al.	395/200.56
5,452,454	9/1995	Basu	395/652
5,475,421	12/1995	Palmer et al.	348/15
5,475,819	12/1995	Miller et al.	395/200.33
5,570,291	10/1996	Dudle et al.	364/468.01
5,612,730	3/1997	Lewis	348/8
5,630,066	5/1997	Gosling	395/200.51
5,721,825	2/1998	Lawson et al.	395/200.33

OTHER PUBLICATIONS

Lowe, Jim; "How Java servlets can replace CGI scripts—for ease, performance & more", *Netscape World*, May 5, 1997 (http://www.cs.berkeley.edu/~padmanab/papers/masters-tr.ps), (http://www.netscapeworld.com/netscapeworld/nw-05-1997/nw-05-bytecode.html).

Gorman, Trisha; "Server-side applets in Java generate developer anticipation", *Netscape World*, Jul. 4, 1996, (http://www.netscapeworld.com/netscapeworld/nw-07-1996/mw-07-jeeves.html).

Padmanabhan, Venkata; "Improving World Wide Web Latency", *University of California at Berkeley Computer Science Division Report*, No. UCB/CSD-95-875, (http://www.cs.berkeley.edu/~padmanab/papers/masters-tr.ps).

Brown, Mark R.; "FastCGI: A High-Performance Gateway Interface", *Fifth International World Wide Web Conference*, May 2, 1996 (http://www.fastcgi.com/kit/doc/www5-api-workshop.html).

Editor; "Control of Dynamic Threads Pool for Concurrent Remote Procedure Calls", *IBM Technical Disclosure Bulletin*, vol. 38, No. 5, May 1995, pp. 199-200.

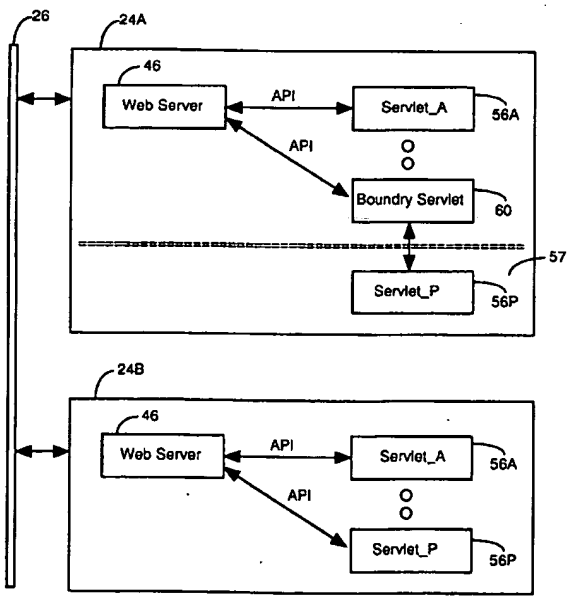
Primary Examiner—Zarni Maung

Attorney, Agent, or Firm—William S. Galliani; Pennie & Edmonds LLP

[57] **ABSTRACT**

A method and apparatus for operating a local server computer of a client-server network includes a technique to receive a request from a client computer of the client-server network. A determination is made whether the request requires dynamically generated information from a servlet object of the client-server network. If so, a specified servlet object corresponding to the request may be uploaded from a remote server computer of the client-server network. The specified servlet object is then executed to obtain dynamically generated information corresponding to the request.

30 Claims, 5 Drawing Sheets



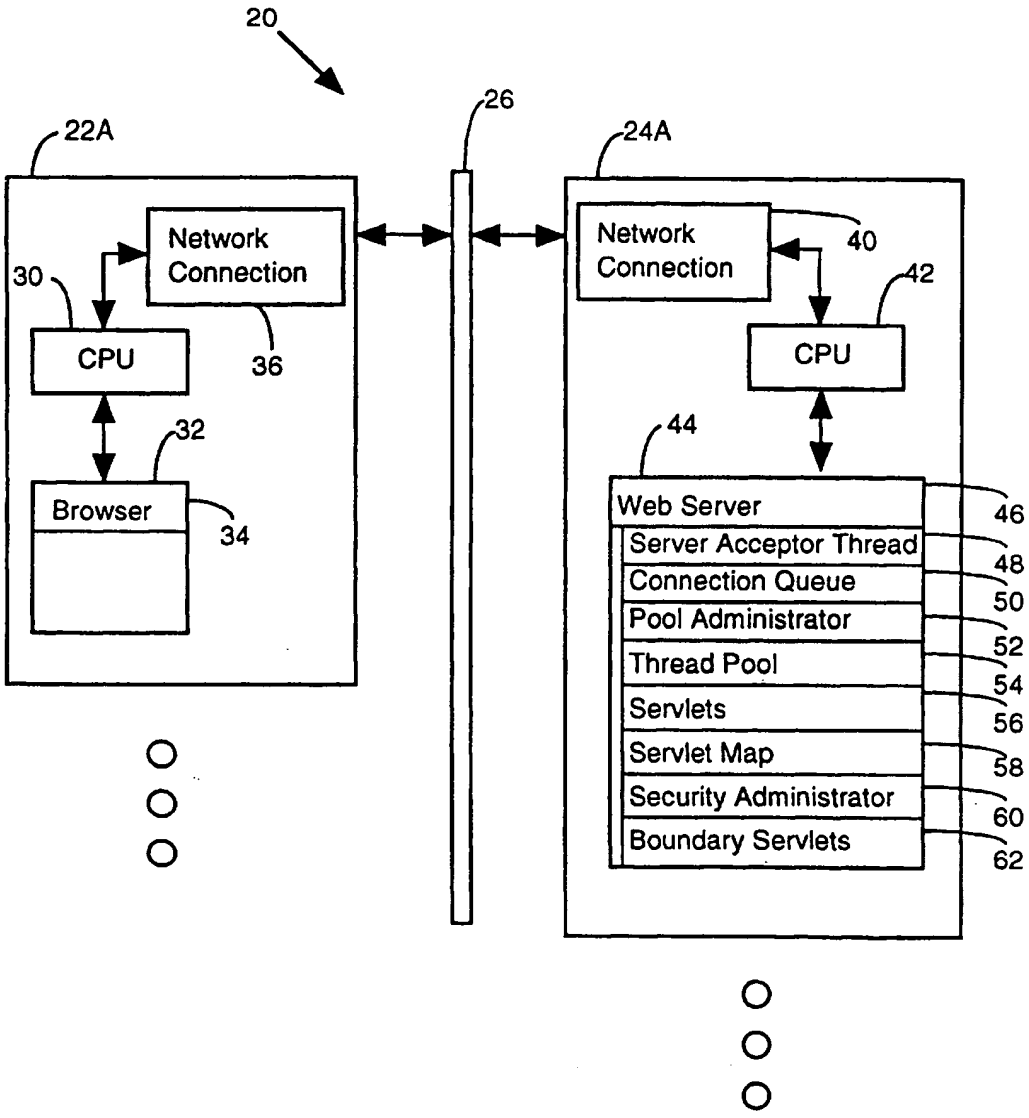

20
22A
26
24A
Network Connection
30
CPU
36
32
Browser
34
Network Connection
40
42
CPU
44
Web Server
46
Server Acceptor Thread
48
Connection Queue
50
Pool Administrator
52
Thread Pool
54
Servlets
56
Servlet Map
58
Security Administrator
60
Boundary Servlets
62

Figure 1

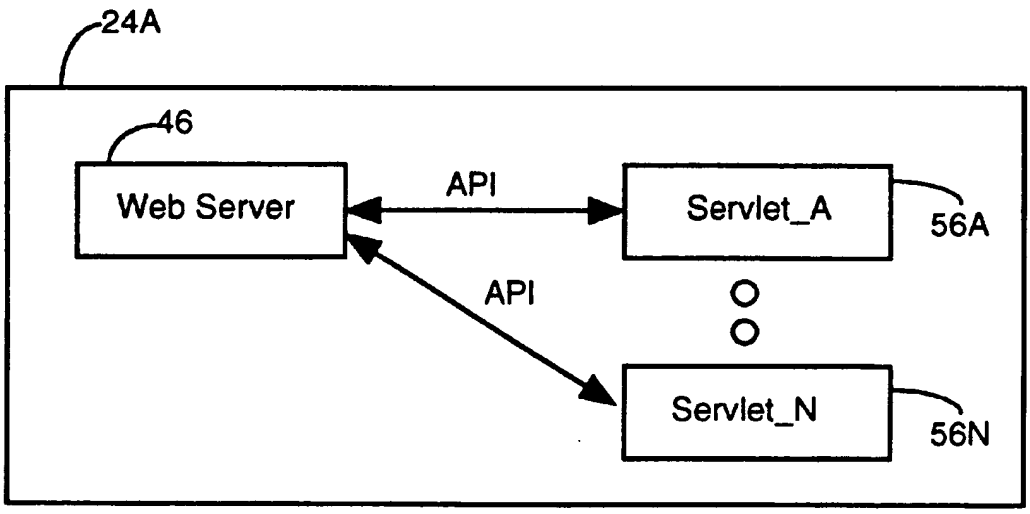


Figure 2

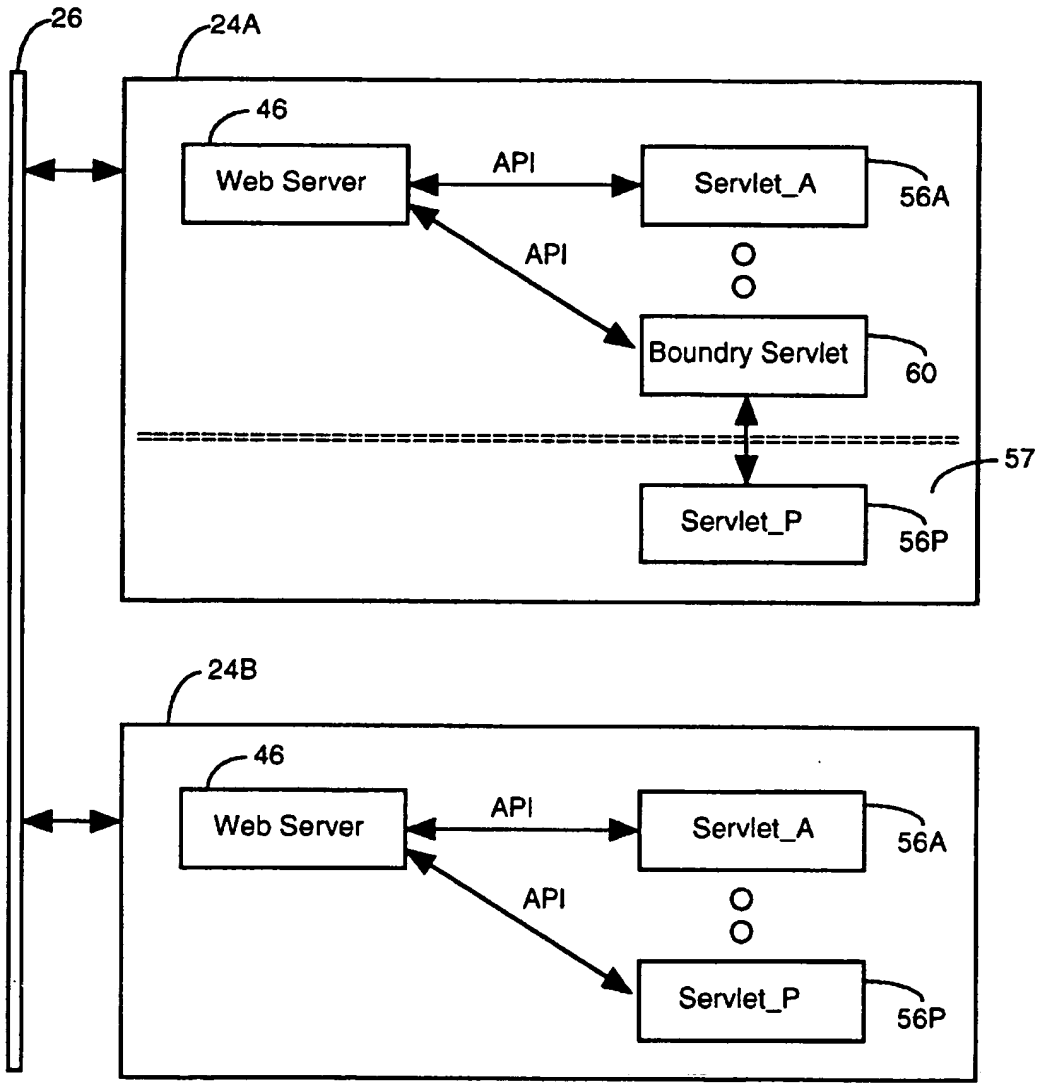

26
24A
46
Web Server
API
Servlet_A
56A
API
Boundry Servlet
60
57
Servlet_P
56P
24B
46
Web Server
API
Servlet_A
56A
API
Servlet_P
56P

Figure 3

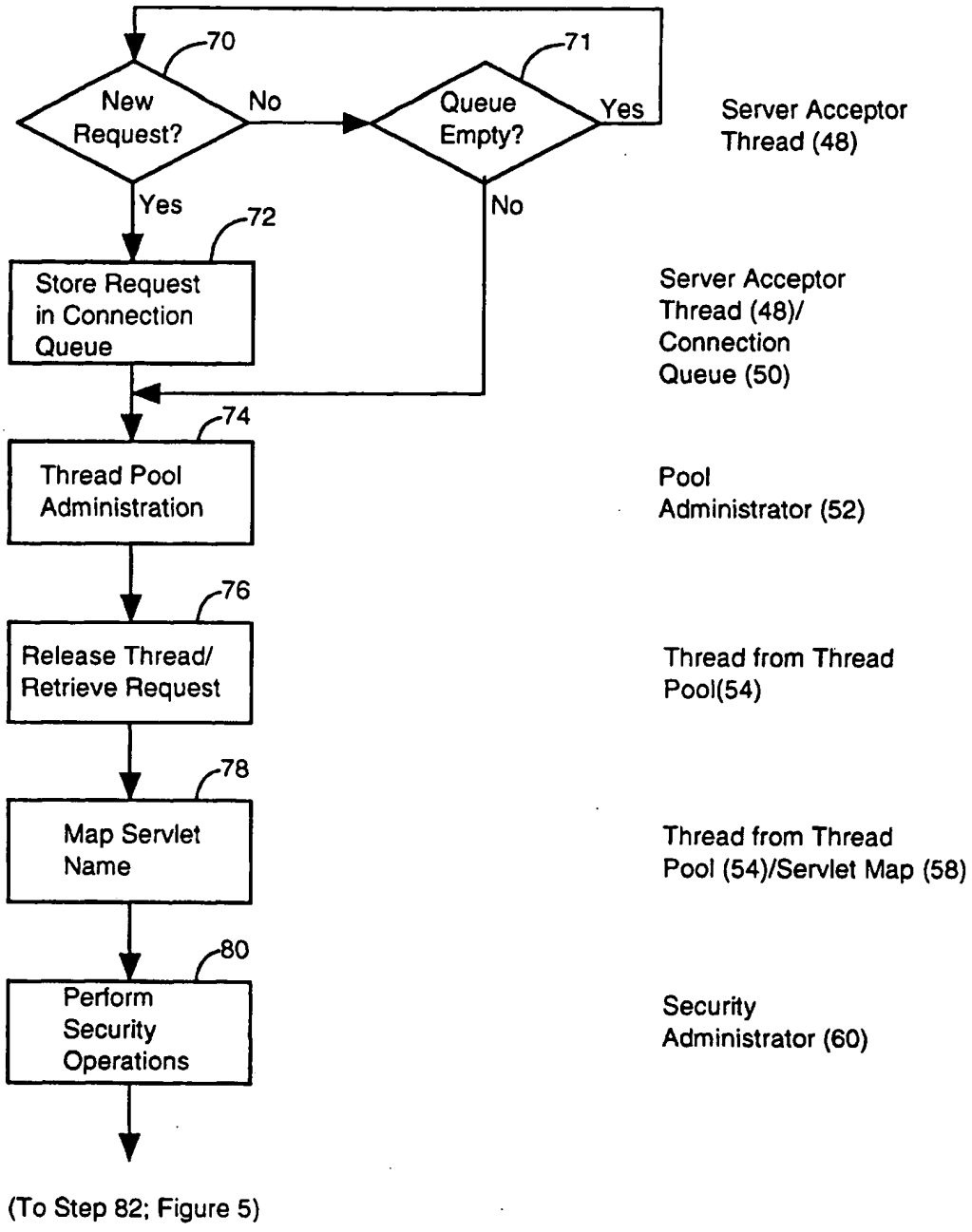

70
New Request?
No
71
Queue Empty?
Yes
Server Acceptor Thread (48)
Yes
No
72
Store Request in Connection Queue
Server Acceptor Thread (48)/ Connection Queue (50)
74
Thread Pool Administration
Pool Administrator (52)
76
Release Thread/ Retrieve Request
Thread from Thread Pool(54)
78
Map Servlet Name
Thread from Thread Pool (54)/Servlet Map (58)
80
Perform Security Operations
Security Administrator (60)
(To Step 82; Figure 5)

Figure 4

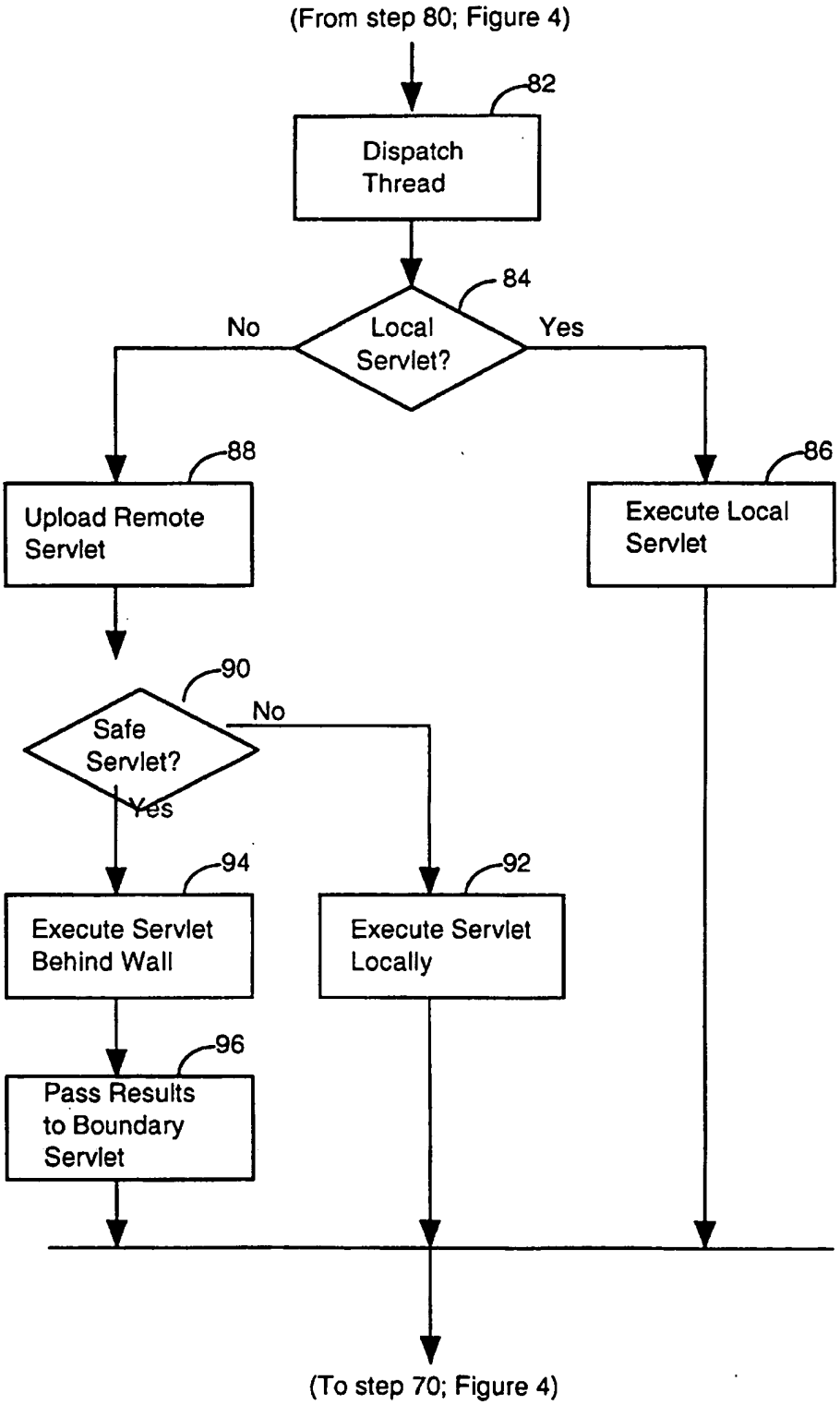

(From step 80; Figure 4)
82
Dispatch Thread
84
Local Servlet?
No
Yes
88
Upload Remote Servlet
86
Execute Local Servlet
90
Safe Servlet?
No
Yes
94
Execute Servlet Behind Wall
92
Execute Servlet Locally
96
Pass Results to Boundary Servlet
(To step 70; Figure 4)

Figure 5

APPARATUS AND METHOD FOR DYNAMICALLY GENERATING INFORMATION WITH SERVER-SIDE SOFTWARE OBJECTS

This application is a continuation-in-part of U.S. Ser. No. 08/756,325 filed Nov. 25, 1996, entitled "Method and System for Facilitating Servlets," now abandoned, which is a continuation of U.S. Ser. No. 08/657,712 filed May 30, 1996, entitled "Method and System for Facilitating Servlets," now abandoned.

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by any one of the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

BRIEF DESCRIPTION OF THE INVENTION

This invention relates generally to exchanging information in a client-server computer environment. More particularly, this invention relates to an improved technique for responding to information requests at a server computer.

BACKGROUND OF THE INVENTION

Client-server computer networks are well known. The most prominent example of a client-server computer network is the World Wide Web of computers. In a client-server computer network, a server computer receives a request for information from a client computer. Web server software operating on the server computer typically retrieves the requested information from a file stored on a permanent storage device and transmits the file over the network to the client computer that requested the information. The web server software is generally not written using an object oriented programming language. Thus, it is not easily extended to provide new functionality. Given the dynamic nature of today's software marketplace, a product's lack of flexibility and extendibility can seriously hinder the marketability of the product.

Current web server software can generate a file dynamically in response to a request from a client computer. Typically, the web server receives the request and then forks a Common Gateway Interface (CGI) process to dynamically create the file. Once the file has been created, the web server software transmits the file back to the client computer. Unfortunately, it is computationally expensive to fork a process each time dynamic information needs to be generated.

In view of the foregoing, it would be highly desirable to provide a web server which dynamically generates information in response to a client computer request, but which does not incur a process start-up expense while generating the dynamic information. Further, it would be highly desirable to provide an object oriented web server environment that is flexible and extendible.

SUMMARY OF THE INVENTION

The invention includes a method and apparatus for operating a local server computer of a client-server network. The invention includes a technique to receive a request from a client computer of the client-server network. A determination is made whether the request requires dynamically generated information from a servlet object of the client-server network. If so, a specified servlet object corresponding to the request may be uploaded from a remote server computer of the client-server network. The specified servlet object is then executed to obtain dynamically generated information corresponding to the request.

The servlet objects of the invention provide an object oriented web server environment which is flexible and extendible. The client-server network of the invention is populated with the servlet objects. The servlet objects operate in a continual loop until invoked. Thus, there is no startup overhead associated with execution of the servlet objects. By observing a common applications program interface, the servlet objects can run in any server environment. A feature of the invention allows untrusted servlet objects to be executed in a secure area, with the dynamically generated information being passed from the secure area into the remaining server environment.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the nature and objects of the invention, reference should be made to the following detailed description taken in conjunction with the accompanying drawings, in which:

FIG. **1** illustrates a client-server computer network in accordance with an embodiment of the invention.

FIG. **2** is a simplified illustration of the interactions between a web server and the servlets of the present invention.

FIG. **3** is a simplified illustration of the interactions between a web server and a servlet loaded from an external server.

FIG. **4** illustrates processing steps associated with a servlet processing routine in accordance with an embodiment of the invention.

FIG. **5** illustrates processing steps associated with a servlet processing routine in accordance with an embodiment of the invention.

Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION OF THE INVENTION

FIG. **1** illustrates a client-server computer network **20** that may be operated in accordance with the present invention. The network **20** includes at least one client computer **22** and at least one server computer **24**. The client computer **22** and the server computer **24** are connected by a transmission channel **26**, which may be any wire or wireless transmission channel.

The client computer **22** is a standard computer including a Central Processing Unit (CPU) **30** connected to a memory (primary and/or secondary) **32**. The memory **32** stores a number of computer programs, including a "browser" **34**. As known in the art, a browser is used to communicate with remote server computers **24** and to visually present the information received from such computers. The client computer **22** establishes network communications through a standard network connection device **36**.

The server computer **24** includes standard server computer components, including a network connection device **40**, a CPU **42**, and a memory (primary and/or secondary) **44**. The memory **44** stores a set of computer programs to implement the processing associated with the invention. The memory **44** stores a web server **46**. The web server **46** may be of the type known in the art, which is modified to include

the additional programs shown in FIG. 1. That is, in an embodiment of the invention, a standard web server 46 is modified to include a server acceptor thread 48, a connection queue 50, a pool administrator 52, a thread pool 54, servlets 56, a servlet map 58, a security administrator 60, and boundary servlets 62.

FIG. 2 is a simplified illustration of a server computer 24 A constructed in accordance with an embodiment of the invention. The figure shows a web server 46 interacting with a set of servlets 56A–56N. In particular, the web server 46 interacts with the servlets through an application program interface (API). As indicated in FIG. 1, the web server 46 and the servlets 56 are stored in memory 44. The web server 46 may be standard web server software that is modified to include the functionality described herein. Each servlet 56 is a piece of software code which is used to dynamically generate information. Each servlet 56 is an instantiated software object waiting to be invoked: Once it is invoked, it dynamically generates information. Note that this technique of dynamically generating information is distinct from the typical process of fetching static information from a permanent storage device. The technique of the invention is similar to a CGI script in the sense that it dynamically generates information. However, unlike a CGI script, a servlet object of the present invention is instantiated at server start-up. Thus, the servlet can be thought of as operating in a continual loop waiting to be executed. Observe that after instantiation there is no computational start-up expense when the servlet is called.

FIG. 3 is a general illustration demonstrating additional features of the invention. FIG. 3 illustrates a local server computer 24A which receives a request from a client computer (not shown) over transmission channel 26. The web server 46 determines that dynamically generated information from a servlet object is required. In this case, the servlet object is not initially on the local server computer 24A, thus it is uploaded by the local server computer 24A from a remote server computer 24B using communication link 26. In the example of FIG. 3, servlet 56P is passed from the remote server computer 24B to the local server computer 24A.

FIG. 3 illustrates another feature of the invention. In particular, it illustrates that the uploaded servlet 56P is executed in a security area 57 of the local server computer 24A. After execution, the results are passed to a boundary servlet 60 in the remaining portion of the local server computer 24A. This security feature allows untrusted servlets to be safely executed.

The foregoing discussion provides a general description of the features and benefits of the invention. Attention now turns to a more detailed description of these features and benefits. The left side of FIG. 4 illustrates processing steps associated with an embodiment of the invention. The right side of FIG. 4 illustrates program components that may be used to execute these operations.

The first processing step shown in FIG. 4 is to determine whether a new request has been received (step 70). As indicated above, a request is a request for information from a client computer 22 to a server computer 24. The operation of a client computer 22 requesting information from a server computer 24 is well known. It is typically performed using a Uniform Resource Locator or URL. A URL specifies a computer and a file. A typical URL is http://SU/123. This URL is an instruction to retrieve the file "123 " from the State University computer "SU" using the Hypertext Transfer Protocol "HTTP".

As shown in FIG. 4, a server acceptor thread 48 is used to process each new request. Preferably, the invention is implemented as a connection-oriented web server with a server acceptor thread that continually loops while accepting requests. Once a request is received, it dispatches the request to a connection queue (step 72). As shown in FIG. 1, the connection queue 50 is formed in the memory of the local server computer 24.

If no new request is received, then a check is made to determine whether the queue is empty (step 71). If the queue is not empty or a new request has been received, processing proceeds to step 74. Step 74 entails thread pool administration operations, which are executed by a pool administrator 52. FIG. 1 illustrates a thread pool 54. The thread pool 54 is a pool of threads that are used for request processing. Individual threads fetch and process requests from the connection queue 50. The pool administrator 52 operates to ensure that there is a thread for each request in the connection queue 50. The pool administrator 52 creates or forks additional threads to handle new requests in the connection queue 50. If a maximum number of threads is reached, the pool administrator 52 blocks new requests from entering the connection queue 50. In such a case, the server computer does not receive new requests. On the other hand, if a thread has been waiting more than a predetermined period of time for a request from the connection queue 50, then the pool administrator 52 will destroy it. Preferably, a new handler thread is created using the buffer space of a destroyed handler thread. In other words, the invention is preferably implemented by using a specific buffer memory space for a thread. When a thread is destroyed, the buffer memory space is cleared, but it is assigned to a new thread. By reusing allocated memory in this manner, this embodiment of the invention minimizes the amount of memory used by the system, especially when compared to systems which allocate and deallocate memory on a per request basis.

After the thread pool administration operations are performed (step 74) a thread retrieves a request from the connection queue (step 76). The thread then maps the request to a servlet name (step 78). The servlet may be specified by a URL, in which case the mapping process is direct. On the other hand, some translation process may be required to identify which servlet will be able to service the request. The mapping operation may be performed in one of the following ways. A server administrator may specify that some kinds of client requests always map to a particular servlet. For example, one which talks to a particular database. A server administrator may specify that part of the client request is the name of the servlet, as found in an administered servlets directory. At many sites, that directory would be shared between servers which share the load of processing for the site's clients. Some servers may be able to automatically invoke servlets to filter the output of other servlets, based on their administrative configuration. For example, particular types of servlet output may trigger post-processing by other servlets, perhaps to perform format conversions. Properly authorized clients may specify the servlet to be invoked, without administrative intervention.

Security operations may also be performed by the thread (step 80). A security administrator 60 may be used to identify trusted and untrusted classes of servlets. The decision to trust a servlet may be established by a set of rules associated with the security administrator 60. For example, the security administrator 60 may decide to trust all local servlets and mistrust all uploaded network servlets. Untrusted servlets are then executed in the security area 57, as shown in FIG. 3. The security administrator 60 may also

be used to determine if the servlet is authorized to perform predetermined risky operations. Security information of this type may be stored in the thread.

JAVA servlets in accordance with the invention provide strong security policy support. This is because all JAVA environments provide a Security Manger which can be used to control whether actions such as network or file access are to be permitted. By default, all servlets are untrusted, and are not allowed to perform operations such as accessing network services or local files. However, servlets "built into" the server, or servlets which have been digitally signed as they were put into JAVA Archive files, may be trusted and granted more permissions by the security manager. A digital signature on executable code indicates that the organization which signed the code "vouches for it" in some sense. Such signatures can't support accountability by themselves, but they do indicate a degree of assurance that may be placed on use of that code. For example, a particular signature from an MIS organization might be required on all code which is granted general access to network services within a corporate intranet. That signature might only be used on code which is strongly believed not to violate particular security policies. Extension APIs in other languages, such as C or scripting languages, can't support such fine grained access controls even if they do allow digital signatures for their code.

After security operations are performed (step **80**), the thread is dispatched (step **82** of FIG. **5**). The dispatch operation entails invoking a servlet so that it generates the requested dynamic information. The dispatch operation is one of two types. A decision is made to determine whether the servlet is local (step **84**). If the servlet is local, then the local servlet is executed (step **86**). This results in the generation of dynamic information that is then processed by the web server **46** in a standard manner. The web server **46** typically passes the information back to the client computer using known techniques. The exchange of information between a servlet and the web server **46** is achieved through an application program interface, which is described below.

If the servlet is not local, then it is uploaded from a remote server **24B** (step **88**). A decision is then made regarding whether the uploaded servlet is safe (step **90**). Recall that the security operation step resulted in the thread acquiring information regarding security parameters for servlets. If there are no security problems associated with the uploaded servlet, then it is executed locally (step **92**). On the other hand, if a security problem is identified, then the servlet is executed in a security area (step **94**). Thereafter, the dynamically generated results are passed to the non-security area (step **96**). A boundary servlet may be used for this purpose. The boundary servlet may be implemented through the use of stubs and subcontracts or through other "fire wall" techniques known in the art. After the servlet is executed, processing returns to step **70** of FIG. **4**.

The operation of the invention has now been fully described. Attention now turns to a more particular discussion of the servlet objects that are used in accordance with the invention and an embodiment of the application program interface used in connection with the servlet objects. As indicated above, the servlet objects are software objects that are used to dynamically generate information. They are instantiated objects that sit in a loop waiting to be invoked. Preferably, they are implemented as object bytecodes in the JAVA**198** programming language. It is well known that the JAVA™ programming language is used to implement "applets" on a client computer. An "applet" is executable JAVA object bytecodes that are used to generate a graphical display on a client computer. The servlets of the present invention are executed on the server side and do not have graphical content.

A servlet is typically instantiated on server startup. In the alternative, the servlet may be instantiated under a predetermined set of conditions or by client invocation. The servlet may be instantiated and executed by using its URL (e.g., http://host/<servlet URL>). The http protocol supports the passing of arguments, thus, arguments may be passed to the servlet (e.g., http://host/<servlet URL>?<arguments>). The properties object is a JAVA programming language properties class which comprises a set of "name:value" pairs. A system administrator can pass arguments to an instantiated HttpServlet object through the properties object. In this way, the system administrator can "customize" an HttpServlet for a particular server at a particular site. For example, the system administrator can pass the Httpservlet object site specific information about the network location of a database which stores documents that will be requested by client processes across the network or the amount of memory available in system buffers which will be used for processing the server administrator.

Once instantiated, a servlet loops until the server is shut off or a destroy method is called on the servlet by the server. Since the servlet operates in a continual loop as it waits for requests to act upon, the server computer avoids the overhead of creating and destroying the servlet between requests to the servlet. In addition, keeping servlets alive between requests allows servlets to pass data and communicate amongst themselves. For example, servlets can maintain data about a user between sessions by the user. This data can be shared among different servlets in order to customize a working environment within which the user works. If servlets were created and destroyed on a per request basis, it would be much more difficult, if not practically impossible, for a servlet to understand the environment within which it runs and utilize this knowledge to provide improved processing capabilities. The server computer can call a destroy method on the servlet when some resource limit in terms of time, memory, etc. is reached.

The servlet application program interface (API) establishes a standard for interfacing servlets with information servers, such as web servers. The servlet API contains methods for initializing a servlet, processing the request, getting servlet information, and destroying the servlet. The servlet API allows platform independent servlets. An example servlet interface is as follows:

```
Servlet interface:
interface HttpServlet {
        Initialize (ServletContext, ServerProperties);
        Service(HttpRequest, HttpResponse);
        Destroy( );
        }
```

The server computer passes objects that implement the "HttpRequest", while the servlet returns an "HttpResponse" object. The "ServletContext" interface is used to exchange information with the server environment. Some of the methods on the "ServletContext" object are "Getserver( )" and "GetServlets( )". "GetServer" returns a pointer to the parent server within which the instantiated Httpservlet runs. Using this pointer, the HttpServlet object can find out information about its parent server. The "GetServlet" method returns pointers to the servlets running on the parent server. The "ServerProperties" interface is used to exchange information regarding specific server properties established by a server administrator.

Servlets support the familiar programming model of accepting requests and generating responses. The following is a simple servlet defining a single method called "service":

```
import java.servlet.*;
public class MyServlet extends GenericServlet {
        public void service (
                ServletRequest request,
                ServletResponse response
        )throws ServletException, IOException
        {
        ...
        }
        ...
}
```

The service method is provided with Request and Response parameters. These parameters encapsulate the data sent by the client, thereby allowing servlets to report status information, such as errors. Servlets normally retrieve most of their parameters through an input stream, and send their responses using an output stream:

ServletInputStream in=request.getInputStream( ):

ServletOutputStream out=response.getOutputStream( );

These input and output streams may be used with data in whatever format is appropriate. For example, an applet and servlet might exchange data using object serialization, HTML, or any number of image formats.

Since servlets are JAVA objects, they have instance-specific data. This means that in effect servlets are independent applications running within servers, without needing the complexity of additional classes (which are required by some alternative server extension APIs). Servlets have access to some servlet-specific configuration data at initialization time. This allows different instances of the same servlet class to be initialized with different data, and be managed as differently named servlets. The data provided at initialization time includes an area where each instance keeps its persistent instance-specific state.

Building upon the previous simple servlet examples, the following program code is an example of a servlet that is used to send Hypertext Markup Language (HTML) text when it is invoked:

```
public class SimpleServlet extends GenericServlet {
public void service(ServletRequest req, ServletResponse res)
        throws ServletException, IOException
    {
            res.setContentType("text/html");
            PrintWriter out = new PrintWriter(res.getOutputStream( ));
            out.println("<HEAD><TITLE> SimpleServlet Output
                    </TITLE></HEAD><BODY>");
        out.println("<h1> SimpleServlet Output </h1>");
        out.println("<P>This is output from SimpleServlet.");
        out.println("</BODY>");
        out.flush( );
    }
    public String getServletInfo( ) {
            return "A simple servlet";
    }
}
```

The following program code is an example of a servlet that uses the finger protocol to query information about users on specified host computers. The query string parameters <tt>user</tt>, <tt>hosts</tt>, and <tt>verbose</tt>can be used to specify the user and hosts to query. The parameter <tt>user</tt>is the user name, <tt>hosts</tt>is a comma-separated list of host names to query, and <tt>verbose</tt>, if specified, will cause verbose output to be generated. For example, <pre>http:/goa/finger.html?user=dac&hosts=eno, doppio&verbose=yes </pre>will request full information about user "dac" on both hosts "eno" and "doppio".

```
public
class FingerServlet extends GenericServlet {
        /*
        *Port number for finger daemon.
        */
        static final int FINGER_PORT = 79;
        /**
        * Handles a single finger request from the client.
        */
        public void service(ServletRequest req, ServletResponse res)
                throws ServletException, IOException
        {
                String user = req.getParameter("user");
                String hosts = req.getParameter("hosts");
                String verbose = req.getParameter("verbose");
                res.setContentType("text/html");
                PrintStream out = new PrintStream(res.getOutputStream( ));
                out.println("<html>");
                out.println("<head><title>Finger Servlet</title></head>");
                out.println("<body>");
                out.println("<h2>Finger results:</h2>");
                outprintln("<pre>");
                if (hosts == null) {
                        finger(out, user, null, "yes".equalsIgnoreCase(verbose));
                } else {
                        StrikingTokenizer st = new StringTokenizer(hosts, ",");
                        while (st.hasMoreTokens( )) {
                                String host = st.nextToken( );
                                out.println("[" + host + "]");
                                try {
                                        finger(out, user, host,
                                "yes".equalsIgnoreCase(verbose));
                                } catch (IOException e) {
                                        out.println(e);
                                }
```

-continued

```
                        out.println( );
                    }
                }
                out.println("</pre>");
                out.println("</body></html>");
        }
        /*
        *       Sends finger output for a user and host to the specified output
        *       stream.
        */
        void finger(OutputStream out, String user, String host, boolean verbose)
                throws IOException
        {
                // open connection to finger daemon
                Socket s;
                if(host == null) {
                        s = new Socket(InetAddress.getLocalHost( ), FINGER_PORT);
                } else {
                        s = new Socket(host, FINGER_PORT);
                }
                // send finger comand
                PrintStream ps = new PrintStream(s.getOutputStream( ));
                if(verbose) {
                        ps.print("/W");
                }
                if(user ! = null) {
                        ps.print(user);
                }
                ps.print("\r\n");
                ps.flush( );
                // copy results to output stream
                InputStream in = s.getInputStream( );
                byte[ ]buf = new byte [512];
                int len;
                while ((len = in.read(buf, 0, buf.length)) ! = -1) {
                        out.write(buf, 0, len);
                }
                s.close( );
        }
}
```

Those skilled in the art will appreciate that servlets which are being used with the HTTP protocol may support any HTTP method, including GET, POST, BEAD, and more. They may redirect requests to other locations, and send HTTP-specific error messages. They can get access to parameters which were passed through standard HTML forms, including the HTTP method to be performed and the URI, which identifies the destination of the request.

As indicated above, one of the biggest performance features of servlets is that they do not require creation of a new process for each request. In most environments, many servlets run in parallel within the same process as the server. When used in such environments with HTTP, servlets provide compelling performance advantages over both the CGI approach and the Fast-CGI approach. This is because servlets have a small computational expense during thread context switches. Since in most environments servlets can handle many client requests each time they are initialized, the cost of the initialization is spread over many methods. All the client requests to that service have the opportunity to share data and communications resources, benefitting more strongly from system caches.

Those skilled in the art will appreciate that the servlets of the invention can be used to dynamically extends Java-enabled servers. The servlets provide a general framework for services built using the request-response paradigm. The servlets can provide secure web-based access to data which is presented using HTML web pages and they can be used for interactively viewing or modifying that data using dynamic web page generation techniques.

The servlets of the invention may be used to provide customized multi-user services for customer bases. The servlets are also flexible enough to support standardized services, such as serving static web pages through the HTTP (or HTTPS) protocols, and proxying services. Since they are used for dynamic extensibility, they may be used in a plug-in style, supporting facilities such as search engines and semi-custom applications, such as web-based order entry or inventory systems.

Although the servlets are preferably written in JAVA, the servlet clients may be written in any language. When servlets are used in the middle tiers of distributed application systems, they can in turn be clients to other services, written in any language.

Those skilled in the art will appreciate that servlets may be used in several modes. The basic mode is at the core of a request/response protocol. In addition, servlets may be specialized to support protocols such as HTTP. In HTTP based applications, servlets are portable, complete, and much more efficient replacement for CGI based extensions. Also, in HTTP applications, servlets may be used with HTML server side includes to dynamically generate part of a web document.

The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be apparent to one skilled in the art that the specific details are not required in order to practice the invention. In other instances, well known circuits and devices are shown in block diagram form in order to avoid unnecessary distraction from the underlying invention. Thus, the foregoing descriptions of specific embodiments of the present invention are presented for purposes of illustration and description. They are not

intended to be exhaustive or to limit the invention to the precise forms disclosed, obviously many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following Claims and their equivalents.

We claim:

1. A method executed by a local server computer under the control of a program, said local server computer including a memory for storing said program, said local server computer forming a portion of a client-server network, said method comprising the steps of:

receiving a request from a client computer of said client-server network;

determining that said request requires dynamically generated information from a servlet object of said client-server network;

uploading from a remote server computer of said client-server network a specified servlet object corresponding to said request; and

executing said specified servlet object to obtain dynamically generated information corresponding to said request.

2. The method of claim 1 further comprising the step of passing dynamically generated information from said specified servlet object to a web server operating on said local server computer, said passing step being facilitated with an application program interface.

3. The method of claim 2 wherein said application programming interface specifies techniques for performing at least one of the following operations: initializing a servlet object, executing a servlet object, and destroying a servlet object.

4. The method of claim 2 wherein said specified servlet object and said application program interface are specified as object bytecodes in the JAVA programming language.

5. The method of claim 2 further comprising the step of sending said dynamically generated information from said web server to said client computer.

6. The method of claim 1 wherein said executing step includes the steps of executing said specified servlet in a security area of said local server computer; and

passing said dynamically generated information from said security area to a non-security area of said local server computer.

7. The method of claim 1 wherein said local server computer stores a plurality of servlet objects, each of said servlet objects continuously operating until invoked in response to a specified request from a client computer.

8. The method of claim 7 wherein said plurality of servlet objects pass data to one another.

9. The method of claim 7 wherein said selected servlet objects of said plurality of servlet objects are instantiated at the start-up of said local server computer.

10. The method of claim 7 wherein selected servlet objects of said plurality of servlet objects are instantiated in response to a demand from said client computer.

11. The method of claim 7 wherein selected servlet objects of said plurality of servlet objects are instantiated in response to an activated servlet URL.

12. The method of claim 11 wherein said servlet HTML includes arguments.

13. The method of claim 1 wherein said receiving step includes the step of storing said request in a connection queue.

14. The method of claim 13 wherein said determining step includes the step of selecting a handler thread from a pool of handler threads to execute said determining step.

15. The method of claim 14 further comprising the step of operating said pool of handler threads by selectively creating a new handler thread and destroying an old handler thread.

16. The method of claim 15 wherein said operating step includes the step of reusing a buffer memory space of said old handler thread for said new handler thread.

17. A computer readable memory that can be used to direct a server computer of a client-server computer network to function in a specified manner, comprising:

a first set of instructions to receive a request from a client computer of said client-server network;

a second set of instructions to determine that said request requires dynamically generated information from a servlet object of said client-server network;

a third set of instructions to upload from a remote server computer of said client-server network a specified servlet object corresponding to said request; and

a fourth set of instructions to execute said specified servlet object to obtain dynamically generated information corresponding to said request.

18. The apparatus of claim 17 further comprising a fifth set of instructions to pass, through an application program interface, dynamically generated information from said specified servlet object to a web server operating on said local server computer.

19. The apparatus of claim 18 further comprising a sixth set of instructions to pass said dynamically generated information from said web server to said client computer.

20. The apparatus of claim 17 further comprising a seventh set of instructions to store a plurality of servlet objects on said server computer, each of said servlet objects continuously operating until invoked in response to a specified request from a client computer.

21. The apparatus of claim 20 wherein said seventh set of instructions include instructions to pass data between said plurality of servlet objects.

22. The apparatus of claim 20 wherein said fifth set of instructions include instructions to pass data between said plurality of servlet objects.

23. A computer readable memory that can be used to direct a server computer of a client-server computer network to function in a specified manner, comprising:

a first set of instructions to receive a request from a client computer of said client-server computer network;

a second set of instructions to determine that said request requires dynamically generated information from a servlet object of said server computer;

a third set of instructions to execute said specified servlet object to obtain dynamically generated information corresponding to said request; and

a fourth set of instructions to pass said dynamically generated information to said client computer.

24. The apparatus of claim 23 wherein said second set of instructions include instructions to interpret a servlet URL corresponding to said request.

25. The apparatus of claim 23 wherein said second set of instructions include instructions to interpret a servlet URL with arguments.

26. The apparatus of claim 23 further comprising a fifth set of instructions to store a plurality of servlet objects on said server computer, each of said servlet objects continuously operating until invoked in response to a specified request from a client computer.

27. A client-server computer network, comprising:

a client computer to generate a request; and

a server computer to determine that said request requires dynamically generated information from a servlet object of said server computer,

execute said specified servlet object to obtain dynamically generated information corresponding to said request, and

pass said dynamically generated information to said client computer.

28. The apparatus of claim 27 further comprising a remote server computer storing a set of servlet objects that can be passed to said server computer.

29. The apparatus of claim 27 wherein said server computer stores a plurality of servlet objects, each of said servlet objects continuously operating until invoked in response to a specified request from said client computer.

30. The apparatus of claim 29 wherein said plurality of servlet objects pass data between themselves.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 5,928,323

DATED : July 27, 1999

INVENTOR(S) : GOSLING et al.

It is certified that an error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 12, line 1, change "HTML" to --URL--.

Signed and Sealed this

Fifteenth Day of February, 2000

Attest:

Q. TODD DICKINSON

Attesting Officer

Commissioner of Patents and Trademarks

US006009416A

United States Patent [19]

Pintsov

[11] Patent Number: 6,009,416

[45] Date of Patent: Dec. 28, 1999

[54] **SYSTEM AND METHOD FOR DETECTION OF ERRORS IN ACCOUNTING FOR POSTAL CHARGES IN CONTROLLED ACCEPTANCE ENVIRONMENT**

[75] Inventor: **Leon A. Pintsov**, West Hartford, Conn.

[73] Assignee: **Pitney Bowes Inc.**, Stamford, Conn.

[21] Appl. No.: **09/052,418**

[22] Filed: **Mar. 31, 1998**

[51] **Int. Cl.**[6] **G06F 17/00**

[52] **U.S. Cl.** **705/410**; 705/406; 364/478.15; 380/23

[58] **Field of Search** 705/1, 401, 406, 705/410; 380/23, 25; 364/478.01, 478.07, 478.08, 478.09, 478.11, 478.12, 478.13, 478.14, 478.15

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,757,537	7/1988	Edelmann et al.	380/51
4,775,246	10/1988	Edelmann et al.	380/23
4,780,828	10/1988	Whisker	364/464.02
4,831,555	5/1989	Sansone et al.	364/519
4,837,701	6/1989	Sansone et al.	364/464.03
4,853,864	8/1989	Hart et al.	364/464.02
4,907,161	3/1990	Sansone et al.	364/464.02
5,448,641	9/1995	Pintsov et al.	380/51
5,612,889	3/1997	Pintsov et al.	364/478.14
5,646,997	7/1997	Barton et al.	380/23
5,675,650	10/1997	Cordery et al.	380/23
5,684,706	11/1997	Harmon et al.	705/406

Primary Examiner—Emanuel Todd Voeltz
Assistant Examiner—Thomas A. Dixon
Attorney, Agent, or Firm—Charles R. Malandra, Jr.; Michael E. Melton

[57] **ABSTRACT**

A mail generation system and method includes means for processing data to generate mail piece information and first secure processing means for securely storing and encrypting mail piece information generated by the data processing means. Means are coupled to the data processing means for physically preparing mail pieces related to the generated mail piece information and for generating information related to the physical preparation of the mail. Second secure processing means securely store and encrypted information generated by the mail preparing means. Means sort the mail pieces and generate information related to the sorting and packaging of the mail pieces. Third secure processing means securely store and encrypt information generated by the mail sorting and packaging means. A part of the software program used to generate the mail piece information can be securely stored. Mail piece information to verify that the software program was employed to generate the mail piece information is encrypted.

8 Claims, 7 Drawing Sheets

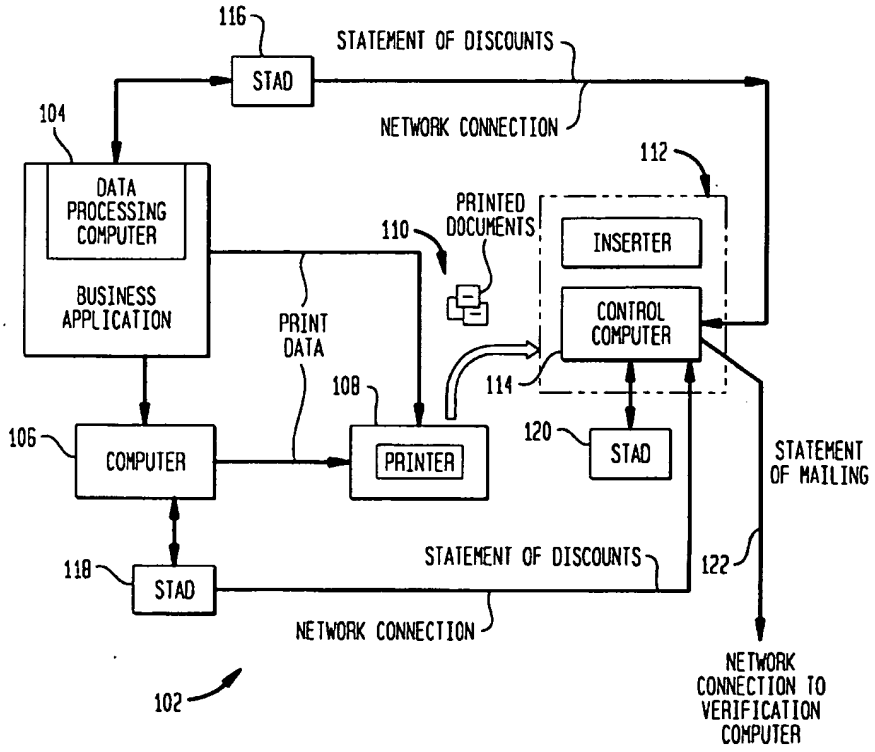


FIG. 1

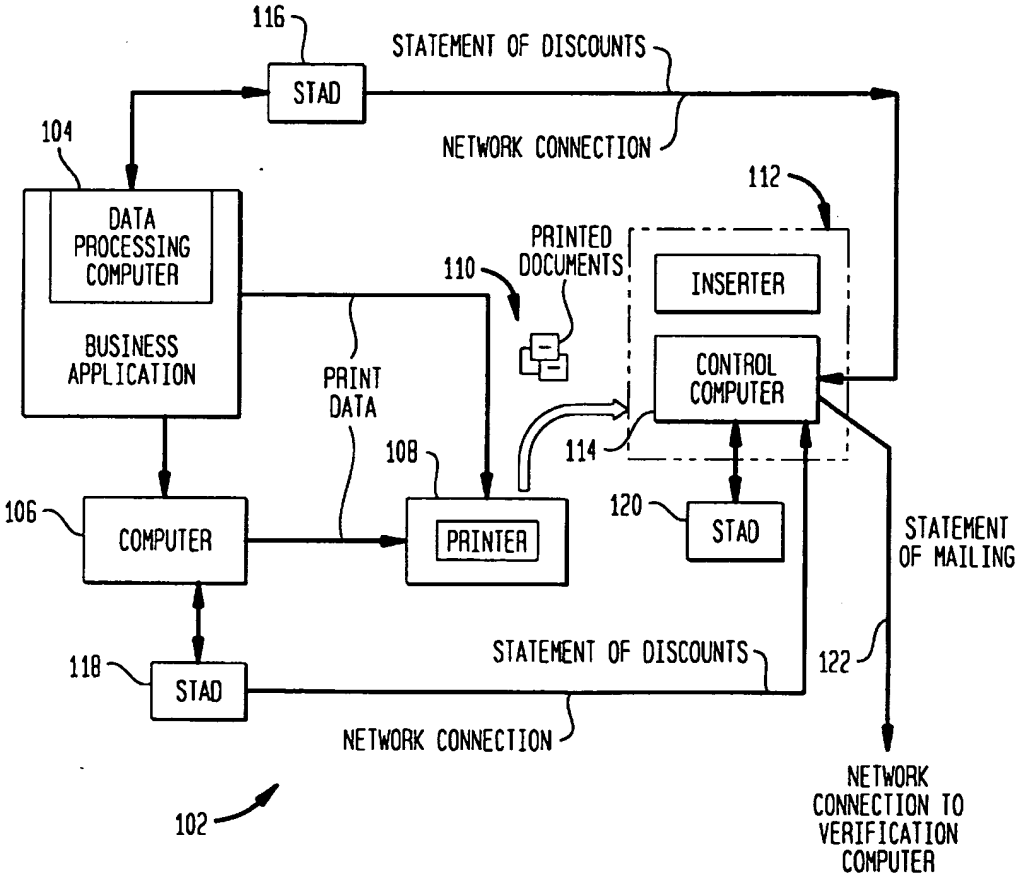

116
STAD
STATEMENT OF DISCOUNTS
NETWORK CONNECTION
104
DATA PROCESSING COMPUTER
BUSINESS APPLICATION
112
PRINTED DOCUMENTS
110
INSERTER
CONTROL COMPUTER
PRINT DATA
108
114
106
COMPUTER
PRINTER
120
STAD
STATEMENT OF MAILING
118
STAD
STATEMENT OF DISCOUNTS
122
NETWORK CONNECTION
NETWORK CONNECTION TO VERIFICATION COMPUTER
102

FIG. 2

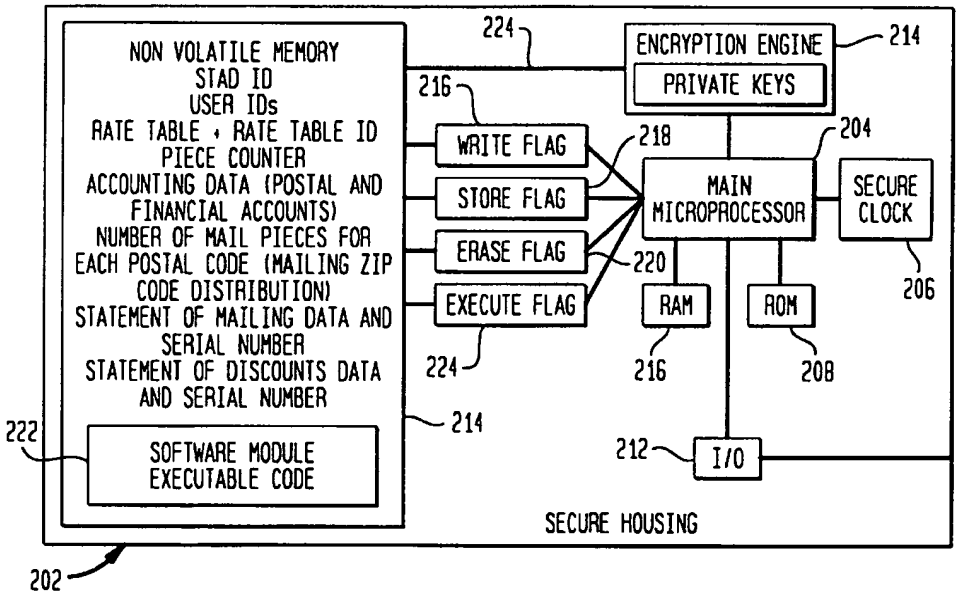

NON VOLATILE MEMORY
STAD ID
USER IDs
RATE TABLE + RATE TABLE ID
PIECE COUNTER
ACCOUNTING DATA (POSTAL AND FINANCIAL ACCOUNTS)
NUMBER OF MAIL PIECES FOR EACH POSTAL CODE (MAILING ZIP CODE DISTRIBUTION)
STATEMENT OF MAILING DATA AND SERIAL NUMBER
STATEMENT OF DISCOUNTS DATA AND SERIAL NUMBER
SOFTWARE MODULE EXECUTABLE CODE
224
ENCRYPTION ENGINE
PRIVATE KEYS
214
216
WRITE FLAG
STORE FLAG
ERASE FLAG
EXECUTE FLAG
218
204
MAIN MICROPROCESSOR
SECURE CLOCK
220
RAM
ROM
206
216
208
224
214
222
212
I/O
SECURE HOUSING
202

FIG. 3

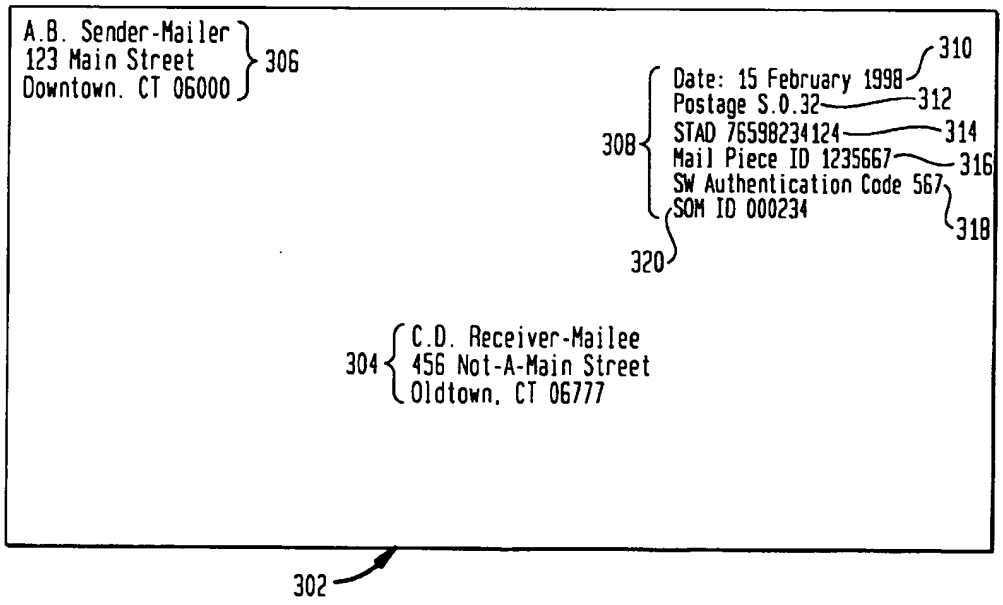
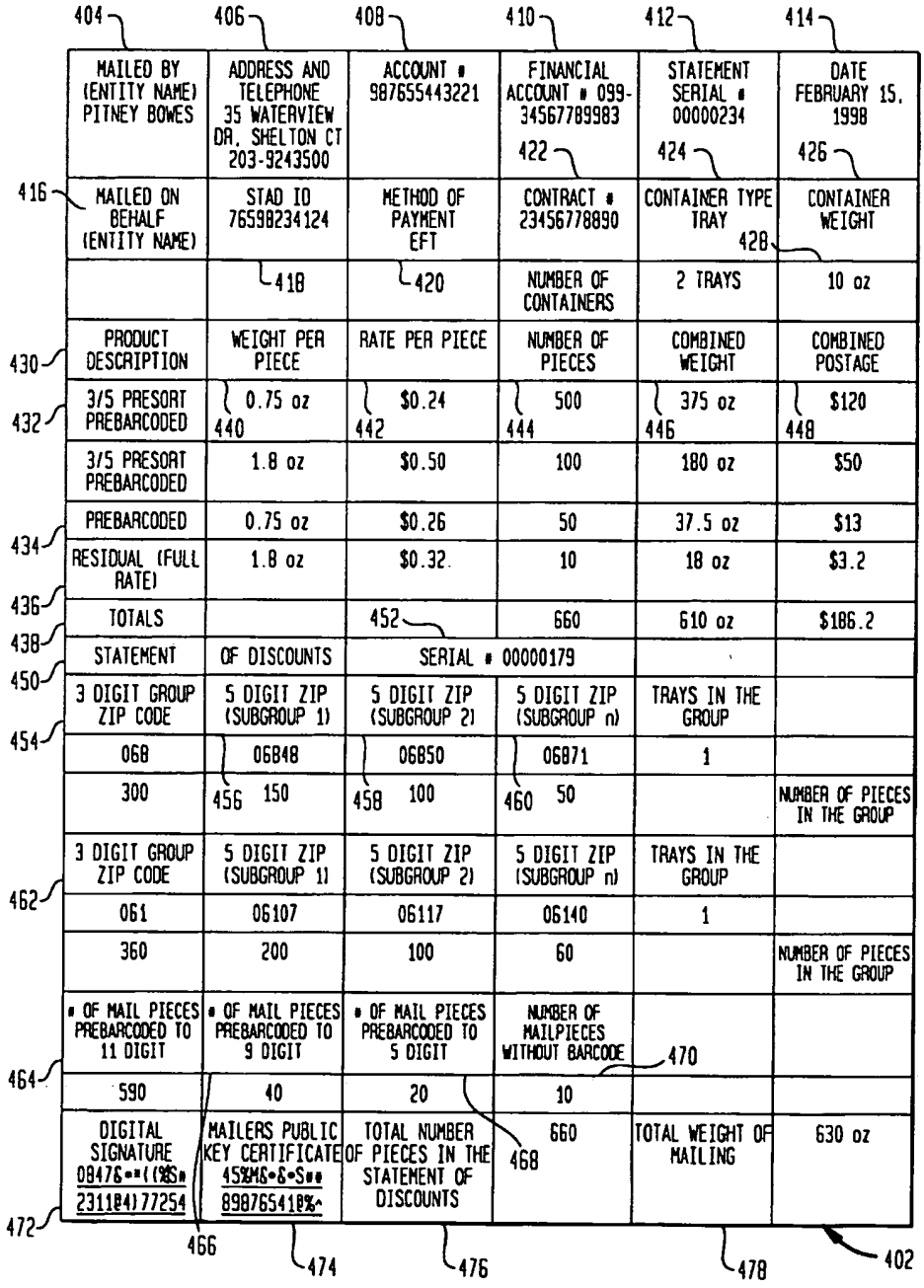

A.B. Sender-Mailer
123 Main Street
Downtown. CT 06000
306
310
Date: 15 February 1998
Postage S.0.32
312
STAD 76598234124
314
308
Mail Piece ID 1235667
316
SW Authentication Code 567
SOM ID 000234
318
320
304
C.D. Receiver-Mailee
456 Not-A-Main Street
Oldtown. CT 06777
302

FIG. 4



404	406	408	410	412	414
MAILED BY (ENTITY NAME) PITNEY BOWES	ADDRESS AND TELEPHONE 35 WATERVIEW DR. SHELTON CT 203-9243500	ACCOUNT # 987655443221	FINANCIAL ACCOUNT # 099-345677899983 422	STATEMENT SERIAL # 00000234 424	DATE FEBRUARY 15, 1998 426
416 MAILED ON BEHALF (ENTITY NAME)	STAD ID 765598234124 418	METHOD OF PAYMENT EFT 420	CONTRACT # 23456778890	CONTAINER TYPE TRAY 428	CONTAINER WEIGHT
			NUMBER OF CONTAINERS	2 TRAYS	10 oz
430 PRODUCT DESCRIPTION	WEIGHT PER PIECE	RATE PER PIECE	NUMBER OF PIECES	COMBINED WEIGHT	COMBINED POSTAGE
432 3/5 PRESORT PREBARCODED	0.75 oz 440	$0.24 442	500 444	375 oz 446	$120 448
3/5 PRESORT PREBARCODED	1.8 oz	$0.50	100	180 oz	$50
434 PREBARCODED	0.75 oz	$0.26	50	37.5 oz	$13
RESIDUAL (FULL RATE)	1.8 oz	$0.32	10	18 oz	$3.2
436 TOTALS		452	660	610 oz	$186.2
438 STATEMENT	OF DISCOUNTS	SERIAL # 00000179			
450 3 DIGIT GROUP ZIP CODE	5 DIGIT ZIP (SUBGROUP 1)	5 DIGIT ZIP (SUBGROUP 2)	5 DIGIT ZIP (SUBGROUP n)	TRAYS IN THE GROUP	
454 068	06848	06850	06871	1	
300	456 150	458 100	460 50		NUMBER OF PIECES IN THE GROUP
3 DIGIT GROUP ZIP CODE	5 DIGIT ZIP (SUBGROUP 1)	5 DIGIT ZIP (SUBGROUP 2)	5 DIGIT ZIP (SUBGROUP n)	TRAYS IN THE GROUP	
462 061	06107	06117	06140	1	
360	200	100	60		NUMBER OF PIECES IN THE GROUP
# OF MAIL PIECES PREBARCODED TO 11 DIGIT	# OF MAIL PIECES PREBARCODED TO 9 DIGIT	# OF MAIL PIECES PREBARCODED TO 5 DIGIT	NUMBER OF MAILPIECES WITHOUT BARCODE 470		
464 590	40	20	10		
DIGITAL SIGNATURE 0847&**((%$# 2311#4)77254 472	MAILERS PUBLIC KEY CERTIFICATE 45%M&*&*S## 89876541@%^ 466 474	TOTAL NUMBER OF PIECES IN THE STATEMENT OF DISCOUNTS 476	660 468	TOTAL WEIGHT OF MAILING 478	630 oz 402

FIG. 5

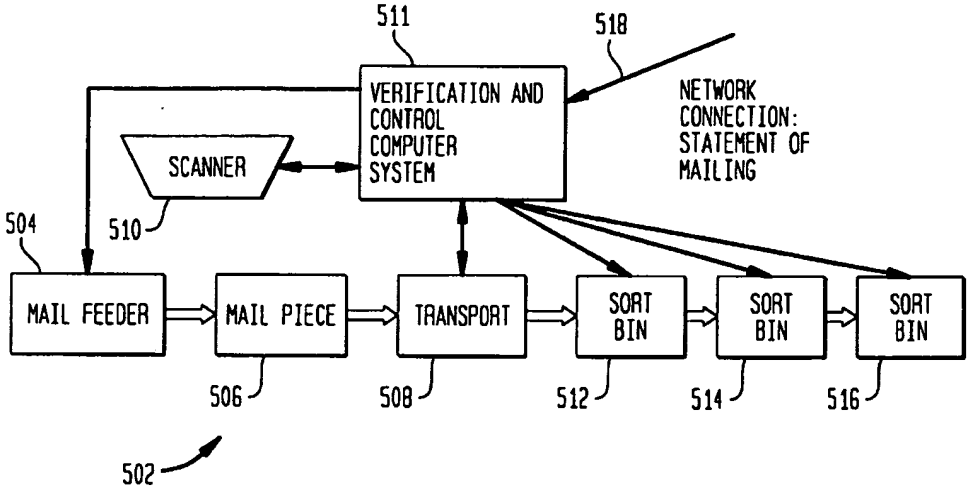

511
518
VERIFICATION AND CONTROL COMPUTER SYSTEM
NETWORK CONNECTION: STATEMENT OF MAILING
SCANNER
510
504
MAIL FEEDER
MAIL PIECE
TRANSPORT
SORT BIN
SORT BIN
SORT BIN
506
508
512
514
516
502

FIG. 6

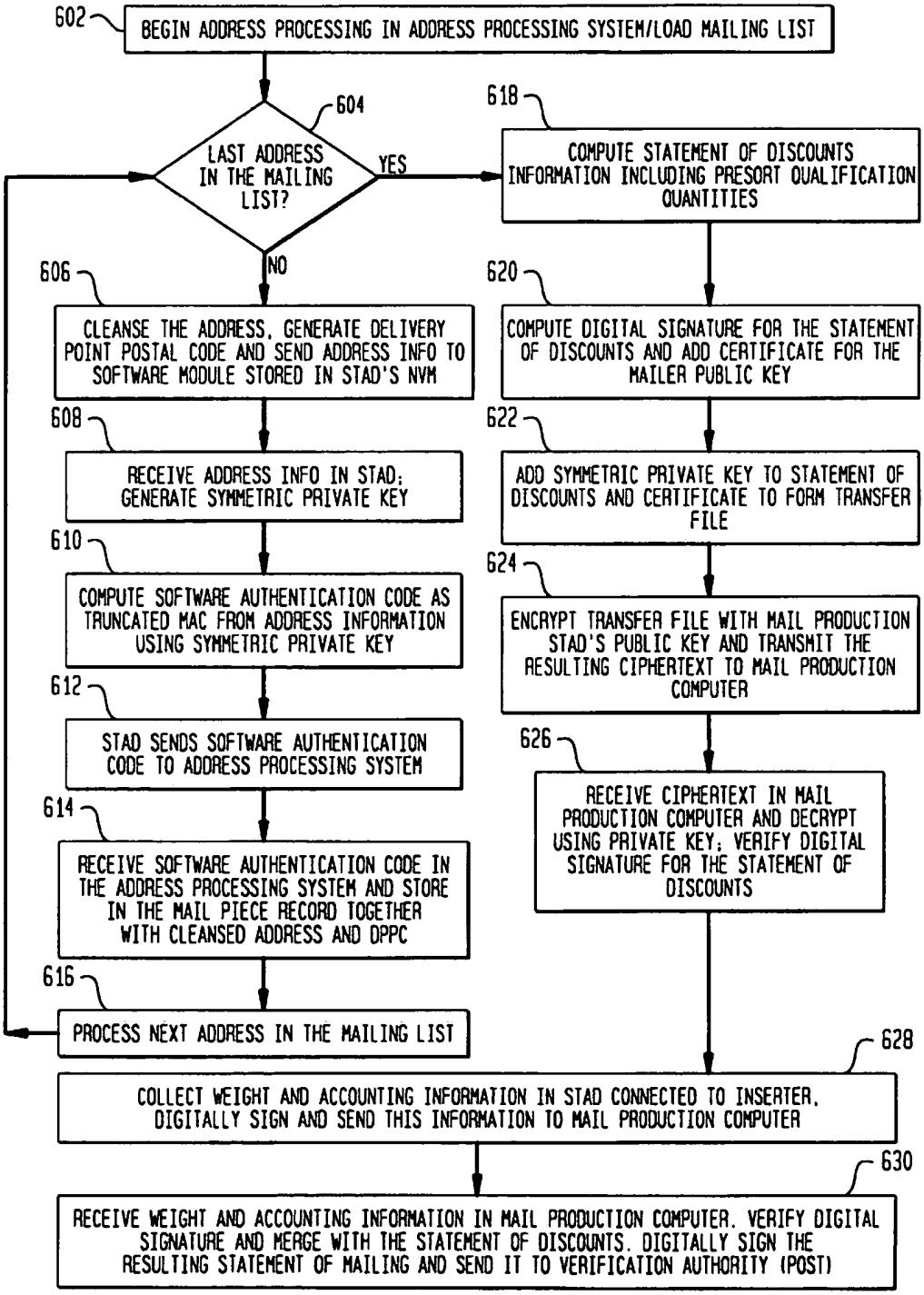
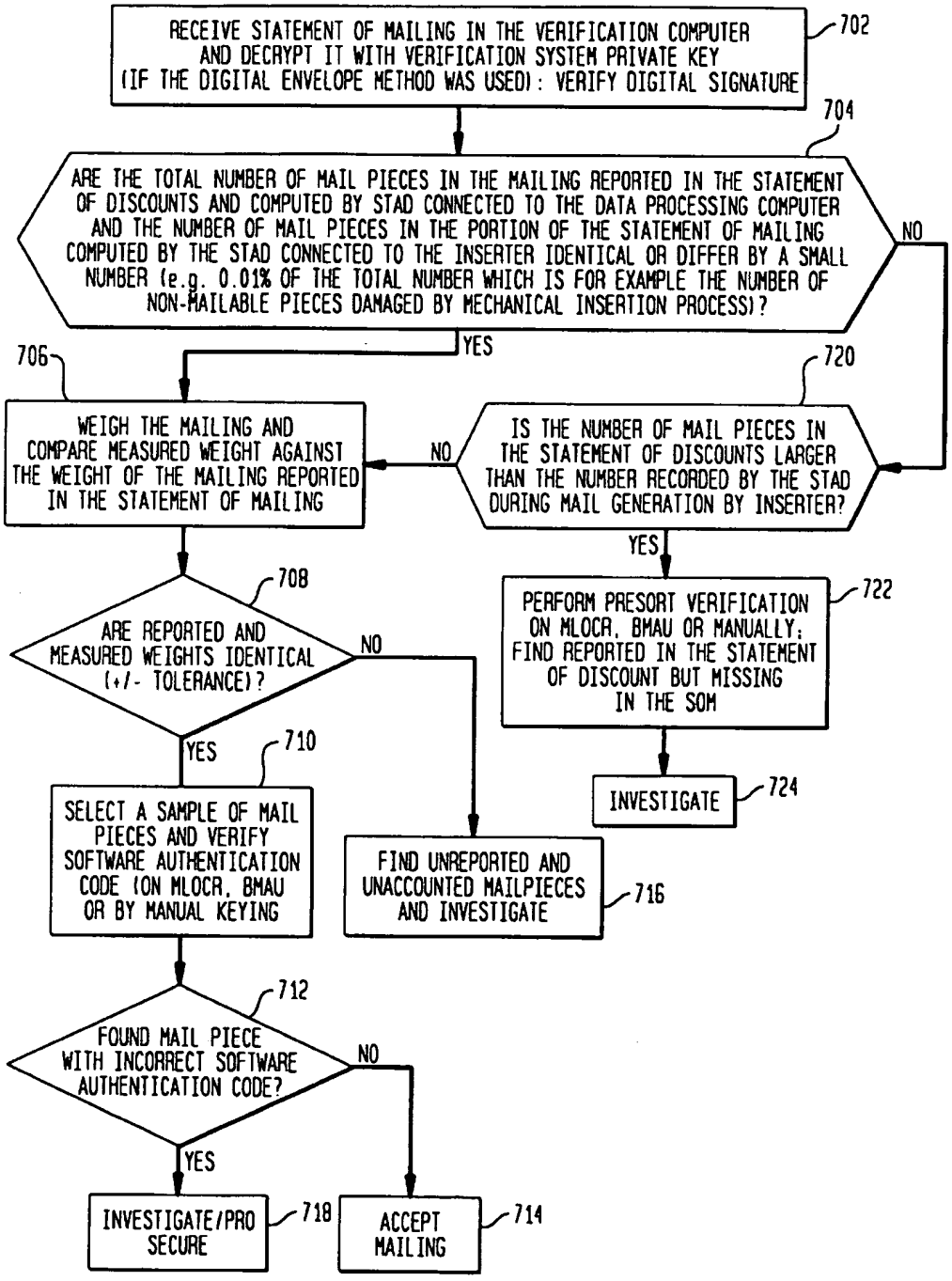

602
BEGIN ADDRESS PROCESSING IN ADDRESS PROCESSING SYSTEM/LOAD MAILING LIST
604
LAST ADDRESS IN THE MAILING LIST?
YES
NO
618
COMPUTE STATEMENT OF DISCOUNTS INFORMATION INCLUDING PRESORT QUALIFICATION QUANTITIES
606
CLEANSE THE ADDRESS, GENERATE DELIVERY POINT POSTAL CODE AND SEND ADDRESS INFO TO SOFTWARE MODULE STORED IN STAD'S NVM
620
COMPUTE DIGITAL SIGNATURE FOR THE STATEMENT OF DISCOUNTS AND ADD CERTIFICATE FOR THE MAILER PUBLIC KEY
608
RECEIVE ADDRESS INFO IN STAD; GENERATE SYMMETRIC PRIVATE KEY
622
ADD SYMMETRIC PRIVATE KEY TO STATEMENT OF DISCOUNTS AND CERTIFICATE TO FORM TRANSFER FILE
610
COMPUTE SOFTWARE AUTHENTICATION CODE AS TRUNCATED MAC FROM ADDRESS INFORMATION USING SYMMETRIC PRIVATE KEY
624
ENCRYPT TRANSFER FILE WITH MAIL PRODUCTION STAD'S PUBLIC KEY AND TRANSMIT THE RESULTING CIPHERTEXT TO MAIL PRODUCTION COMPUTER
612
STAD SENDS SOFTWARE AUTHENTICATION CODE TO ADDRESS PROCESSING SYSTEM
626
RECEIVE CIPHERTEXT IN MAIL PRODUCTION COMPUTER AND DECRYPT USING PRIVATE KEY; VERIFY DIGITAL SIGNATURE FOR THE STATEMENT OF DISCOUNTS
614
RECEIVE SOFTWARE AUTHENTICATION CODE IN THE ADDRESS PROCESSING SYSTEM AND STORE IN THE MAIL PIECE RECORD TOGETHER WITH CLEANSED ADDRESS AND DPPC
616
PROCESS NEXT ADDRESS IN THE MAILING LIST
628
COLLECT WEIGHT AND ACCOUNTING INFORMATION IN STAD CONNECTED TO INSERTER, DIGITALLY SIGN AND SEND THIS INFORMATION TO MAIL PRODUCTION COMPUTER
630
RECEIVE WEIGHT AND ACCOUNTING INFORMATION IN MAIL PRODUCTION COMPUTER. VERIFY DIGITAL SIGNATURE AND MERGE WITH THE STATEMENT OF DISCOUNTS. DIGITALLY SIGN THE RESULTING STATEMENT OF MAILING AND SEND IT TO VERIFICATION AUTHORITY (POST)

FIG. 7


RECEIVE STATEMENT OF MAILING IN THE VERIFICATION COMPUTER AND DECRYPT IT WITH VERIFICATION SYSTEM PRIVATE KEY (IF THE DIGITAL ENVELOPE METHOD WAS USED): VERIFY DIGITAL SIGNATURE
702
704
ARE THE TOTAL NUMBER OF MAIL PIECES IN THE MAILING REPORTED IN THE STATEMENT OF DISCOUNTS AND COMPUTED BY STAD CONNECTED TO THE DATA PROCESSING COMPUTER AND THE NUMBER OF MAIL PIECES IN THE PORTION OF THE STATEMENT OF MAILING COMPUTED BY THE STAD CONNECTED TO THE INSERTER IDENTICAL OR DIFFER BY A SMALL NUMBER (e.g. 0.01% OF THE TOTAL NUMBER WHICH IS FOR EXAMPLE THE NUMBER OF NON-MAILABLE PIECES DAMAGED BY MECHANICAL INSERTION PROCESS)?
NO
YES
706
WEIGH THE MAILING AND COMPARE MEASURED WEIGHT AGAINST THE WEIGHT OF THE MAILING REPORTED IN THE STATEMENT OF MAILING
NO
720
IS THE NUMBER OF MAIL PIECES IN THE STATEMENT OF DISCOUNTS LARGER THAN THE NUMBER RECORDED BY THE STAD DURING MAIL GENERATION BY INSERTER?
YES
708
ARE REPORTED AND MEASURED WEIGHTS IDENTICAL (+/- TOLERANCE)?
NO
YES
710
SELECT A SAMPLE OF MAIL PIECES AND VERIFY SOFTWARE AUTHENTICATION CODE (ON MLOCR, BMAU OR BY MANUAL KEYING
FIND UNREPORTED AND UNACCOUNTED MAILPIECES AND INVESTIGATE
716
PERFORM PRESORT VERIFICATION ON MLOCR, BMAU OR MANUALLY; FIND REPORTED IN THE STATEMENT OF DISCOUNT BUT MISSING IN THE SOM
722
INVESTIGATE
724
712
FOUND MAIL PIECE WITH INCORRECT SOFTWARE AUTHENTICATION CODE?
NO
YES
INVESTIGATE/PRO SECURE
718
ACCEPT MAILING
714

SYSTEM AND METHOD FOR DETECTION OF ERRORS IN ACCOUNTING FOR POSTAL CHARGES IN CONTROLLED ACCEPTANCE ENVIRONMENT

FIELD OF THE INVENTION

The present invention pertains to mail payment and evidencing systems and, more particularly, to a mail payment and evidencing system which is adapted to be employed with a batch of mail prepared by a mailer and processed by a carrier as part of the mail distribution process.

BACKGROUND OF THE INVENTION

Various methods have been developed for payment of carrier services. These payment methods include postage stamps which are individually applied to each mailpiece and metered imprints which are also individually applied to each mailpiece. Additionally, other systems have been developed such as permit mail where a carrier issues a permit allowing certain types of mailing and manifest systems wherein mail is manifested and delivered to a carrier service along with the manifest.

In a mail production environment, where large batches of mail are produced, each of the above payment methods involves compromises between ease of use and security for the payment of postage to the carrier service. Stamped mail requires costly printing of stamps by the carrier service, as well as costly control and revenue accounting for the stamps. Moreover, the utilization of stamps as a payment method provides little information to the carrier service related to the cost associated with operating any particular facility or any particular class of mail delivery service provided. Additionally, the utilization of stamps particularly in a large mail production environment, does not easily accommodate multiple rate mailings. Mechanical dispensing of stamps is slow and prone to malfunction. The labor and time involved in purchasing of stamps by the mailer is costly, and security is limited due to theft, of stamps and reused or "washing" of stamps.

Traditional metered mail provides a significant level of security for the carrier service. However, in high volume production mail environment variable weight mailings may require multiple meters to achieve high throughput speeds and mechanical malfunctions may frequently occur for high volumes of mail printed by meters with mechanical printing mechanisms.

Many of these problems have been alleviated with the advent of new electronic postage meters, particularly postage meters which are adapted to print with digital printing technologies. Enhanced security has been obtained with postage meters with digital printing through the use of encrypted indicias. The encrypted indicias employ a digital token which is encrypted data that authenticates the value and other information imprinted on the mailpiece. Examples of systems for generating and using digital tokens are described in U.S. Pat. No. 4,757,537 for SYSTEM FOR DETECTING UNACCOUNTED FOR PRINTING IN A VALUE PRINTING SYSTEM; U.S. Pat. No. 4,831,555 for UNSECURED POSTAGE APPLYING SYSTEM; and, U.S. Pat. No. 4,775,246 for SYSTEM FOR DETECTING UNACCOUNTED FOR PRINTING IN A VALUE PRINTING SYSTEM. Because the digital token incorporates encrypted data including postage value, altering of the printed postage revenue and the postage revenue block is detectable by an appropriate verification procedure.

Moreover, systems have been proposed for postal payment with verifiable integrity to detect attempts to interfere with the rating process for the postage amount to be imprinted as opposed to interference with the resulting printed postage value. In this connection, reference is made to U.S. Pat. No. 5,448,641 for POSTAL RATING SYSTEM WITH A VERIFIABLE INTEGRITY.

Both permit mail and manifest mail systems, as well as related contract mail systems, usually have no evidence of postage payment on individual mailpieces and require complex and extensive acceptance procedures and associated documentation. These systems are very complex, time consuming and inaccurate for the carrier service in administering and accepting mail. Moreover, the funds security of the system is vulnerable since it is open to undetectable collusion. Once permit mail has been accepted into the carrier mail delivery system, it is extremely difficult to determine whether the mail has been paid for. Furthermore, because of the various techniques used for payment adjustments, a significant loss of revenue or over payment by either the carrier or the mailer, as the case may be, is possible since payment is verified only by a sampling method. In addition, systems of this type are very complex for the mailer, are error prone and require extensive documentation. Further, the risk of overpayment by the mailer or the requirement to redo the documentation and mail due to adjustments exists in these systems. Additionally, the systems of this type involve time consuming costly acceptance procedures. Moreover, for certain of these permit payment systems, preprinted envelopes must be maintained in inventory.

An improved manifest system has been proposed, for example, as set forth in U.S. Pat. No. 4,907,161 for BATCH MAILING SYSTEM, U.S. Pat. No. 4,837,701 for MAIL PROCESSING SYSTEM WITH MULTIPLE WORK STATIONS; U.S. Pat. No. 4,853,864 for MAILING SYSTEM HAVING POSTAL FUNDS MANAGEMENT; U.S. Pat. No. 4,780,828 for MAILING SYSTEM WITH RANDOM SAMPLING OF POSTAGE; and U.S. Pat. No. 5,675,650 for CONTROLLED ACCEPTANCE MAIL PAYMENT AND EVIDENCING SYSTEM.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an improved postage payment and evidencing system.

It is a further object of the present invention to provide an effective controlled acceptance process for such mail that includes improved flexibility for the mailer in creating mail and a high level of security for payment and evidencing of appropriate carrier service.

It is yet a further objective of the present invention to employ a system for batch mail along with verification procedures in the creation and physical preparation of the mail.

A mail generation system embodying the present invention includes means for processing data to generate mail piece information and first secure processing means for securely storing and encrypting mail piece information generated by the data processing means. Means are coupled to the data processing means for physically preparing mail pieces related to the generated mail piece information and for generating information related to the physical preparation of the mail. Second secure processing means securely store and encrypted information generated by the mail preparing means. Means sort the mail pieces and generate information related to the sorting and packaging of the mail pieces. Third secure processing means securely store and

encrypt information generated by the mail sorting and packaging means.

A mail generation method for embodying the present invention includes processing data to generate mail piece information and securely storing and encrypting mail piece information generated by the data processing. Mail pieces related to the generated mail piece information are physically prepared and information related to the physical preparation of the mail generated. Information generated by the mail preparing is securely stored and encrypted. Information related to the sorting and packaging of the mail pieces is generated and the information generated by the mail sorting and packaging is securely stored and encrypted.

In accordance with a feature of the invention, a method for mail generation includes processing data to generate mail piece information and securely storing a part of the software program used to generate the mail piece information. Mail piece information to verify that the software program was employed to generate the mail piece information is encrypted.

In accordance with yet another feature of the present invention, a method for mail generation includes processing data to generate mail piece information and securely storing and encrypting mail piece information generated by the data processing. Mail pieces related to the generated mail piece information are physically prepared and information related to the physical preparation of the mail is generated. Information generated by the mail preparing is securely stored and encrypted. A comparison is made of the securely stored and encrypted mail piece information generated by the data processing and the securely stored and encrypted information generated by said mail preparing means.

In accordance with still another aspect of the invention, mail may be physically inspected for consistency with the securely stored and encrypted mail piece information generated by the data processing and the securely stored and encrypted information generated by the mail preparing means.

BRIEF DESCRIPTION OF THE DRAWINGS

Reference is now made to the following Figures wherein like reference numerals designate similar elements in the various views and in which:

FIG. **1** is a diagrammatic depiction of a batch mail generation system employing the present invention;

FIG. **2** is a secure trusted accounting device suitable for use in the system shown in FIG. **1**.

FIG. **3** is a mail piece created in accordance with aspects of the present invention.

FIG. **4** is a secure statement of mailing including statement discounts generated by the system shown in FIG. **1**.

FIG. **5** is a verification system for mail pieces created by the system shown in FIG. **1**.

FIG. **6** is a flow chart for the process of generation of secured statement of mailing including statement of discounts; and

FIG. **7** is a flow chart for the process of verification of the secure statement of mailing including statement of discounts.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

General Background

Physical mail is the lifeblood of the mail communication system. The mail communication system remains the only universal means of communication between businesses and customers, e.g. households as well as between households.

Billing is a classical example of a critical business function accomplished through mail communication system. For example, a large utility company such as a telephone company produces and sends on a regular basis (typically monthly) bills to its customers. From information point of view each bill is composed of billing data (such as account number, itemized charges and totals, due date etc.) and the delivery address where the bill must be sent by mail. The billing data is a message or a document.

Production of mail by large mailers is a complex process frequently involving several stages. The delivery address (or simply address if there is no confusion with origination address) and message data are normally created, processed and maintained in a Data Processing environment where powerful main frame or mini computers process large amount of data required to generate mail. Almost all information processing functions for mail creation takes place in this environment including addresses verification, presorting, creation of the information for mail prebarcoding and generation of machine-readable codes for mail assembly machines also known as inserters. If the mail composition data (i.e. a set of parameters sufficient to compute postal rate for each mail piece) known at this stage postal charges are also computed and Statement of Mailing or manifest information is created. These are physical or electronic documents containing among other things summary of postal charges based on mail rating parameters such as weight, presort level, prebarcoding, postal zone etc. Then mailing components are printed by a high speed printing systems. These components are sheets of paper with message information, address information and machine readable assembly instructions. After the printing process, printed components are brought into mail production facilities where they are merged with other materials and assembled into finished mail pieces. During this process, postal charges may be computed by an insertion machine (if it was not possible to do so during the Data Processing stage) and imprinted on individual mail items or summarized in a Statement of Mailing or both. Typically, the postal charges computed during the mail production phase when the mail composition is not known at the time of printing of the message and the address/control code bearing documents.

All mailers which produce sizable amounts of mail wish to take advantage of worksharing discounts whenever possible. These are frequently mail charge discounts for presorting and/or prebarcoding discounts. If the number of mail pieces produced or geographical distribution of delivery addresses are not sufficient to qualify for presort discounts, mailers frequently physically merge their mailings with other mailings and presort resulting mailings on production mail sorters similar to ones used by postal operators. Alternatively, mailer may choose to bring the nonqualified portion of their mailings to a service company for merging and presorting with mailings from other companies in exchange for a portion of postal discount. Finally, mail is delivered for controlled acceptance into a postal or other facility where accuracy of the charges computed by the mailer may be verified by postal employees before mail is accepted for distribution. The verification may be of a sample of the mail. In this environment errors, intentional or accidental, are frequent. In USA the incorrectly claimed discounts may be large and even exceed hundreds of millions of dollars annually. It has been discovered, that the problem lies not with the actual physical presort or the quality of bar codes, but with the accounting for such presort

or prebarcoding. The reason for this phenomenon is that mailers are not interested in submitting physically incorrectly presorted mailings because this will affect the quality and timeliness of delivery of their mail thus defeating the purpose of mail communication. However, unscrupulous mailers are very much interested in presenting incorrect accounts to maximize their discounts. The problem is aggravated by the fact that being caught with the incorrect accounting such mailers facing no risk. They are required to pay additional charges assessed by postal acceptance clerks when discovered, but they can try to present incorrectly accounted for mailings again and again. Methods proposed to solve the problem by "certifying" presort/prebarcoding software. These approaches, in principle, have severe limitations since they provide no binding link between physical mail and software used to produce such mail. The unscrupulous mailer can simply use different than "certified" software for producing actual mail or use "certified" software to processes some fictitious addresses artificially added to the real mailing list, which would never make it into actual mailing. In either case "certified" software accomplishes very little in achieving the goal of revenue protection.

In a U.S. Pat. No. 5,675,650 assigned to the same assignee as the present invention an effective mechanism for verifying the number of mail pieces accounted by a secure trusted accounting device has been already described. This mechanism enables the verification authority to find any discrepancy between the reported and accounted and the actual numbers of mail pieces in the mailing, thus enabling quick and effective detection of mail pieces which were not accounted for but present in the mailing. This is the case of the outright stealing of full postage for unreported number of mail pieces. The present case describes extension of this concept to a more subtle case of stolen postal discounts.

System Overview

It has been discovered that the accounting for presorted and/or prebarcoded pieces can be done in conjunction with address processing in a secure manner. This means that all the information required to compute postal discounts is normally available at the time of the mailing list processing and can be supplied to a secure trusted accounting device (STAD). The STAD is electronic hardware and associated software where such information is securely stored. The information in STAD can not be changed once it is entered in STAD, but can be completely erased if required. Upon completion of mailing list processing the STAD contains in its non-volatile memory (NVM) a complete record of the number mail pieces to be produced together with their respective postal codes. This information can be digitally signed and submitted in computerized form directly to the postal acceptance unit where postal computer can verify the digital signature thus making sure that the information was not changed in transit and so the postal computer would have a computerized record of exactly the same information as was submitted by mailer's address processing software to the STAD. The information file produced by STAD and communicated to the Post (verification authority) is a Statement of Mailing, which may include complete set of information regarding discounts, applied by the mailer. We call this part of the Statement of Mailing the Statement of Discounts. The Statement of Mailing is digitally signed and can be communicated to the Post together with the public key certificate signed by the Post or other certification authority. It can be also communicated in the form of digital envelope (see, for example, page 20 Book I Business Description in the publication Secure Electronic Transaction (SET) Specification published Jun. 17, 1996, by Master Card and Visa). This may be particularly advantageous since it will allow to transport the entire Statement of Mailing encrypted using a session symmetric key encrypted with the Postal authority public key. It also allows to include in the message the symmetric secret key which was used to compute digital tokens imprinted on individual mail pieces to provide secure linkage to software used for address processing. This delivers a very effective, and simple, key management system.

From the Statement of Discounts postal computer can then compute presort qualification profile, being, for example, the number of pieces that belong to 3 digit postal code level, 5 digit level etc. together with the estimated number of trays to each 3 digit level and the number of 5 digit postal code bundles in each tray labeled with the corresponding 3 digit postal code. This information can be compared during the acceptance process with the composition of physical mail presented for acceptance using an appropriate sampling procedure. Any discrepancy between the STAD records and the records obtained as a result of physical examination of mailing in the total number of pieces which is estimated based on the total weight as described in U.S. Pat. No. 5,675,650, the entire specification of which is hereby incorporated by reference, in the number of pieces that were addressed to a given postal code etc. would not only indicate fraud but present a very substantial evidence of fraud sufficient for prosecution.

One modification of the present invention allows to securely link every mail piece with its Statement of Discounts. This is done by imprinting or labeling every mail piece with an encrypted number obtained from the delivery address information for the piece, a piece unique identification number and the Statement of Mailing ID. The encrypted number (more appropriately known as the ciphertext or digital token) can be in the form of a truncated Message Authentication Code or obtained by any other appropriate cryptographic primitive which provides for source authentication and data integrity (see Handbook of Applied Cryptography, CRC Press 1997). If such a secure link is implemented it provides a mechanism for proving deliberate fraudulent activities.

A very important benefit of the present invention is the ability to provide evidence of fraud and thus generates a serious deterrence effect. Unscrupulous mailer would have a serious problem claiming an innocent processing error and would have a difficult time in trying to defraud postal authority by a similar method again. The basic method described here can be extended to a number of other alternatives such as to the mail presorted by mailers using physical sorting (not computerized sorting). In this case each physical mail sorter is equipped with STAD that keeps record of presort activities. If the final mailing to be submitted for acceptance by the Post was produced or presorted by several sorters or inserters, the aggregate Statement of Mailing including Statement of Discounts can be combined from such statements produced by individual STADs attached to each machine computer controller. This can be done by a computing device such as a PC equipped with another STAD. In this case individual statements submitted to such a PC digitally signed (or MACed). The PC verifies each signature, assures the authenticity and integrity of data, and then merges all records together and digitally signs the aggregate statement.

It should be expressly noted that in the case when mailer's Electronic Data Processing and Mail Production facilities

are not co located two separate STADs can be used in conjunction with Data (Address) Processing and Mail Assembly. At the end of address processing activity the Statement of Discounts is digitally signed and can be transmitted to a computing device in mail production facility. This transmission can be done via a network such as LAN, WAN or public network such as Internet. In the latter case the Statement of Discounts can be encrypted using for example the digital envelope mode mentioned above. Alternatively, the Statement of Discounts can be physically transferred using magnetic or optical storage device such as floppy diskette or CD ROM. In either case the computing device in the mail production facility is capable of receiving and interpreting the Statement of Discounts. At the end of the mail production run, when the STAD connected to mail generation system, for example, an inserter contains all other data needed to form a Statement of Mailing the two files (Statement of Discounts and mail generation file containing weights and postage by category and other information as described below) are merged. We refer to the combined file as the Statement of Mailing. It is digitally signed and sent to the verification authority (Post) with the digital signature, signature and certificate or in the form of the digital envelope (if privacy protection is required).

The Statement of Mailing contains as a minimum all the information about mailing and its generation process needed to verify that the accounting process was performed properly and all the charges are correctly computed by the mailer's equipment. Alternatively, if as a result of the verification process verification authority determines (by taking physical measurements of the mailing and performing tests and comparing the results of such tests and measurements with the secure information in the Statement of Mailing) that accounting was not done properly, the verification authority will be in the possession of evidence of deliberate fraudulent activities on the part of the mailer. The process allows for noted above generalization when several mail assembly machines (inserters) or several Electronic Data Processing computers are involved in the preparation of the mailing.

It has been also discovered that a certain modification of STAD can provide a proof that specific software program was used to produce given mailing. This is particularly important in the case when postal authorities insist that mailers use "certified" software program for address processing, such as CASS certified software in the USA. In order to produce the evidence that a mail piece was generated using a specific software program the program and the STAD are modified in the following manner. A certain part of the software program, which must be executed for each mail piece, is implemented in firmware and stored within the non-volatile memory of the STAD. Then, when this software program processes mailing list, it must send information (address information) needed to execute the portion stored within the STAD to the STAD where information for software authentication is generated and send back to the main software program for printed inclusion in the information that will be on the mail piece. This authenticating information can be, for example, digital token computed by truncation of a MAC or it could be a digital signature. The authentication is established by the fact that this authenticating information can be generated only upon accessing a secret (hardware protected) key. Implementing address processing software this way forces the address processing computation to access STAD, which in turn then can keep accurate and trusted accounting records. The verification authority can verify the digital token using address information on the mail piece and a secret (or matching public) key shared with the STAD connected to the address processing computer in the mailer's facility and responsible for mail accounting. Thus, the presence of information such as, for example, digital token (truncated MAC) on the mail piece constitutes a proof that a specific software (organized as it is described above) was used to generate the mail piece. It should be noted that the just described methodology can be used for authentication of any software that was used during mail generation process, not only address processing software. For that matter, more generally the described methodology is equally useful when there is a need to ascertain that a certain piece of software was used in generating a certain document which bears evidence of such use. However, the detailed description given below deals only with the address processing software as the preferred embodiment for the most important function in the mail production process.

It has also been discovered that the verification process can be automated by keeping track of mail pieces form the given mailing during physical sortation process by the postal processing equipment such as multi line optical character recognition (MLOCR) sorterer. Alternatively, the verification process can be performed automatically by a Bulk Mail Acceptance Unit (BMAU). The BMAU is a machine used by the United States Postal Service to verify presort qualification by feeding onto a transport a sample of mail or entire mailing; reading addresses and keeping track of the number of mail pieces having certain postal codes. In this functionality, the BMAU is not different than MLOCR.

In addition, the method of present invention can be adopted for use with a special purpose computing system utilized to intercept print files on their way from data processing computer to a printer. Such is the case when main processing software residing for example on a mainframe computer is difficult to modify to extract certain information important for physical mail generation. One such computing system for intercepting and processing print stream is produced by the assignee of the present invention and is known as StreamWeaver®. These and other modifications (some presented below) are entirely within the spirit of present invention.

System Structure and Operation

Reference is now made to FIG. **1**. A mail generation system **102** includes a data processing computer **104** having business application software which is employed to create a mailing. The data processing computer **104** may be connected to a second computer **106** adapted to run a software program for modifying an original print file to be an enhanced print file, which is sent to printer **108**. One suitable software program for changing an original print file to an enhanced print file is the StreamWeaver® to provide print stream processing software marketed by Pitney Bowes Inc. The printer **108** generates a series of printed documents **110** which are further processed by an inserter system **112** having a control computer **114**.

Three secure trusted accounting devices are provided in the system. A first secure trusted accounting device **116** is connected between the data processing computer **104** and the inserter control computer **114**. A second secure trusted accounting device **118** is connected between the print enhanced file computer **106** and the control computer **114**. A third secure trusted accounting device **120** is connected directly to the inserter control computer **114**.

One form of secure trusted accounting device hardware is manufactured by Chrysalis-ITS and is known as the Luna Encryption and Digital Signature Token Device.

It should be recognized that the architecture and the number of secure trusted accounting devices is a matter of choice. The secure trusted accounting device **116** provides a statement of discounts based on the information supplied directly by the data processing computer **104**. Similarly, the secure trusted accounting device **118** also provides a statement of discounts based directly on the information provided by the computer **106**. This information, which is redundant, is supplied to the control computer **114**. A selection may be made to use one or the other of the secure trusted accounting devices **116** and **118** unless there is unique information available only to one and not the other of the secure trusted accounting devices. Secure trusted accounting device **120** provides information concerning the operation of the physical preparation of the mail by the inserter system **112**. It should be noted that the inserter system **112** merely by way of example and can be other equipment involved in the physical preparation and processing of the mail, such as mailing machines, sorters, fully integrated mail generation systems, which includes data processing, packaging, and any other system involved in the physical preparation and processing of the mail.

A statement of mailing, which includes the statement of discounts, is provided to a verification computer through a network connection.

Reference is now made to FIG. **2**. The secure trusted accounting device **202** includes a main microprocessor **204** having a secure clock **206**, a read-only memory (ROM) **208**, random access memory (RAM) **210** and an input/output (I/O) connection **212**.

An encryption engine **214** has private keys securely stored. A flagging system is provided for the computer so that information can be written into the non-volatile memory **214** and can be erased from the non-volatile memory **214**, but cannot be modified once written into the non-volatile memory **214**. The flagging system involves a write flag **216** to enable writing into the non-volatile memory when the store flag **218** is made active. An erase flag **220** is provided to erase information from the non-volatile memory.

The non-volatile memory **214** contains various information useful in processing the mail. This includes the secure trusted accounting device identification, the user identification, the rate table and rate table identification, a piece counter, accounting data and postal and financial accounts information, number of mail pieces for each postal code (mailing ZIP code distribution), statement of mailing data and serial number, and statement of discount data and serial number.

A software module is also provided with executable code at **222**. This software module executable code is a software which is fetched by the main microprocessor to operate as a executable code for a software routine that resides outside of the secure trusted accounting device **202**. This executable code is enabled when an execution execute flag **224** is made active.

It should be recognized that the secure trusted accounting device is housed within a secure tamper-proof housing which may leave telltale signs of attempts to comprise the physical security of the device and have other security features to provide device protection, such as secure connection between the encryption engine and the non-volatile memory shown at **224**. Other secure forms of protection may also be employed.

Reference is now made to FIG. **3**. A mailpiece **302** includes a destination address at **304** and a sender address at **306**. Various information relevant to processing the mail is provided at **308**. This includes the date of mailing at **310**, the postage amount for the mailpiece at **312**, the identification of the secure trusted accounting device which processed the mail at **314**, and a mailpiece identification at **316**.

A software authentication code is provided at **318**. This is a digital token which provides evidence of the fact that the software module executable code **222** was utilized in the preparation and processing of the mail. Finally, a statement of mailing identification code is printed at **320**. This ties the specific mailpiece to a specific piece of mailing document. The digital token may include as part of its input the statement of mailing identification number, which protects the integrity of the information on the mailpiece generally shown at **308**.

It should be recognized that the organization of the printing of the information on the mailpiece is a matter of design choice and can be modified to meet various needs. It can be printed in barcode form to facilitate machine reading of the mailpiece and facilitate automated processing. Various additional information can be included on the mailpiece, depending on the nature of the information desired by the verification authority in processing the mail to provide the integrity desired.

Reference is now made to FIG. **4**. A statement of mailing **402** includes various information relating to the mail created by the system shown in FIG. **1**. The statement of mailing includes the name of the mailer at **404**, the address and telephone number of the mailer at **406**, the internal account number of the mailer at **408**, the banking or financial account number of the mailer at **410**, the statement of mailing serial number at **412**, and the date that the statement of mailing was prepared at **414**. Additional information i provided as to the name of the party on behalf whom the mailing has been prepared, if applicable, at **416** and the secure trusted accounting device identification at **418**. The method of payment is set forth at **420** and the contract number associated with the type of mailing at **422**. This could be, for example, the various contracts that mailers have with the postal services for delivery services related to different categories of mail. The container type, here shown as trays, is noted at **424** as well as the container weight at **426**. The actual weight is shown at **428** as the weight of the cardboard tray in which the mail is stacked. Four different categories of mail are shown under the product description at **430**. These include three/five digit presorted, pre-barcoded (that is, the mail is first sorted to three digit presort and, within each presort, further presorted to five digits.) at **434**, residual at full rate at **436** with the totals being shown at **438**. Within each product description, information is provided as to the weight per piece at **440**, the rate at **442**, the number of pieces at **444**, and the combined weight at **446**. The combined postage is shown at **448**.

A statement of discounts with serial number is shown at **450**. This serial number **452** may be the same as the statement of mailing serial number **412** or may be unique to the statement of discounts itself and related to the statement of mailing. At **454**, further information as to the three digit zip code "068" is shown with 300 pieces. This breaks down as shown in the five digit zip sub-group **1**, **2**, through n, **456**, **458** and **460** with the number of pieces in each five digit zip code sub-group. This information **454–460** is again repeated in area **462** for a different three digit zip code sub-group "061". The number of mailpieces pre-barcoded to eleven digits at **464**, nine digits at **466**, five digits at **468** and without barcodes at **470** is provided. The number of mailpieces in each of these various categories **464–470** is also shown. A digital signature for the statement of mailing is provided at

472 and the mailer's public key certificate is also shown at 474. Finally, the total number of pieces in the statement of discounts is provided at 476 as 660 pieces having a total weight at 478 of 630 ounces.

It should be expressly noted that this statement of mailing may be communicated electronically between the mailer and the carrier system or any trusted third party involved in the processing of the mail. Additionally, the statement of mailing may be printed for physical inclusion with the batch of mail being provided to the carrier service.

Reference is now made to FIG. 5. A mail verification system 502 includes a mixed mail feeder 504, which feeds various mailpieces 506 to a transport 508. A scanner 510 scans the mailpieces as they are transported by transport 508. The transport 508 feeds the mailpieces under the control of the verification and control computer system 511 into a plurality of sort bins 512, 514 and 516. The sortation is based on information obtained via scanning at 510, which information is provided to the verification and control computer 511.

The statement of mailing is provided via the network connection 518 to the verification and control computer system 511. By obtaining the statement of mailing, the verification and control computer system compares the information obtained by the electronic copy of the statement of mailing with the information obtained from scanning the physical mailpieces. This allows verification that the mailing is consistent with the statement of mailing. Alternatively, if it is not consistent, a suitable investigation can be implemented.

Reference is now made to FIG. 6. A mailing list is loaded into the system at 602 to begin processing of the information necessary to generate the mailing. A determination is made at 604 whether the address is the last address in the mailing list. If it is not, the mail processing process continues with the address cleansing and generation of delivery bar code postal code at 606. At 606, additionally, the address information is sent to the software module stored in the secure trusted accounting device's non-volatile memory. Address information in the secure trusted accounting device is received and a symmetric private key is generated at 608. A software authentication code is computed at 610. This code may be a truncated message authentication code (MAC) from address information using symmetric private keys. The secure trusted accounting device sends the software authentication code to the address processing system at 612 and the software authentication code is received in the address processing system at 614. This is stored in the mailpiece record together with the cleansed address and delivery point postal code. At this point, the next address in the mailing list is processed at 616.

When the last address in the mailing list is reached, the statements of discounts is computed at 618, including a presort qualification quantities. This computation is performed in the secure trusted accounting device. A digital signature for the statement of discounts is computed and a certificate for the mailer's public key added at 620. Thereafter, the symmetric private key is added to the statement of discounts and certificate to form a transfer file at 622. The transfer file is encrypted with the mail production secure trusted accounting device's public key and the resulting cipher text is transmitted to the mail production computer at 624.

The cipher text is received in the mail production computer and decrypted using the private key at 626. At this point, the digital signature of the statement of discounts is verified. The weight and accounting information in the secure trusted accounting device is collected and connected to an inserter or other mail processing equipment and digitally signed and transmitted to the mail production computer at 628. At 630, the weight and accounting information is received in the mail production computer and the digital signature is verified. The statement of discounts is merged. The resulting statement of mailing is digitally signed and transmitted to the verification authority, such as a postal authority.

Reference is now made to FIG. 7. The statement of mailing is received at the verification computer at 702 and is decrypted with its verification system private key. The digital signature is then verified. Alternatively, the statement of mailing can be decrypted and verified using the public key certificate appended to the statement of mailing.

At 704, consistency is determined between the secure trusted accounting devices connected to the data processing computer and the inserter. If they are identical or differ by a small number (any number acceptable to the postal authorities), the process may proceed. Where the consistency is acceptable, the measured weight is compared with the weight reported in the statement of mailing at 706. A determination is made at 708 whether the measured and reported weights are identical or within tolerances. If they are within tolerances, a sample of the mailpieces are selected at 710 and the software authentication code is verified. This may be on a MLOCR or BMAU or by manual keying, as determined by the verification facility. A determination is made at 712 whether the mailpieces have a correct or incorrect authentication code. If the mail has the correct authentication code, the mail is accepted at 714 for entry into the mail processing stream. If a determination was made at 708 or 712 that the weights were not within tolerances or the authentication code was incorrect, an investigation is initiated at 716 and/or 718, as the case may be.

Where at 704 an inconsistency is found between the various secure trusted accounting devices, a determination is made at 720 if the number of mailpieces in the statement of discounts is larger than the number recorded by the secure trusted accounting device during the mail generation by the inserter. If this is not the case, the process continues at 706, as previously described.

If, however, the number of mailpieces in the statement of discounts is larger than the number recorded by the secure trusted accounting device during the mail generation by the inserter, presort and verification is performed at 722 by the MLOCR, BMAU or manually, as desired. In such a case, a determination is made to find the missing mailpieces which have been reported in the statement of discounts but are missing in the statement of mailing. As appropriate, an investigation is initiated at 724. This may develop potential evidence of fraud on the part of an unscrupulous mailer.

While the present invention has been disclosed and described with reference to the disclosed embodiments thereof, it will be apparent, as noted above, that variations and modifications may be made.

What is claimed:

1. A mail generation system comprising:

means for processing data to generate mail piece information;

first secure processing means for securely storing and encrypting mail piece information generated by said processing means;

means coupled to said data processing means for physically preparing mail pieces related to said generated

mail piece information and for generating information related to the physical preparation of said mail;

second secure processing means for securely storing and encrypting information generated by said mail preparing means;

means for sorting said mail pieces and for generating information related to said sorting and packaging of said mail pieces; and,

third secure processing means for securely storing and encrypting information generated by said mail sorting and packaging means.

2. A method for mail generation comprising the steps of:

processing data to generate mail piece information;

securely storing and encrypting mail piece information generated by said data processing;

physically preparing mail pieces related to said generated mail piece information and generating information related to the physical preparation of said mail;

securely storing and encrypting information generated by said mail preparing;

generating information related to said sorting and packaging of said mail pieces; and,

securely storing and encrypting information generated by said mail sorting and packaging.

3. A mail generation system comprising:

means for processing data to generate mail piece information;

secure processing means for securely storing and encrypting mail piece information generated by said processing means;

means coupled to said data processing means for physically preparing mail pieces related to said generated mail piece information and for generating information related to the physical preparation of said mail; and,

second secure processing means for securely storing and encrypting information generated by said mail preparing means.

4. A mail generation system as defined in claim 3 wherein said mail piece information which is stored and encrypted relates to information upon which postal processing charges are computed.

5. A method for mail generation comprising the steps of:

processing data to generate mail piece information;

securely storing a part of the software program used to generate said mail piece information;

encrypting mail piece information to verify that said software program was employed to generate said mail piece information; and,

physically preparing mail pieces related to said generated mail piece information.

6. A method for mail generation comprising the steps of:

processing data to generate mail piece information;

securely storing and encrypting mail piece information generated by said data processing;

physically preparing mail pieces related to said generated mail piece information and generating information related to the physical preparation of said mail;

securely storing and encrypting information generated by said mail preparing;

comparing said securely stored and encrypted mail piece information generated by said data processing and said securely stored and encrypted information generated by said mail preparing means.

7. A method for mail generation as defined in claim 6 comprising the further step of physically inspecting said mail.

8. A method for mail generation as defined in claim 6 comprising the further step of physically inspecting said mail for consistency with said securely stored and encrypted mail piece information generated by said data processing and said securely stored and encrypted information generated by said mail preparing means.

* * * * *

United States Patent [19]

Schumacher

[11] **Patent Number: 5,058,030**

[45] **Date of Patent: Oct. 15, 1991**

[54] **OPTIMIZING MAIL PROCESSING BY MATCHING PUBLISHER AND INSERTER ENTITIES**

[75] Inventor: **Karl H. Schumacher**, Westport, Conn.

[73] Assignee: **Pitney Bowes Inc.**, Stamford, Conn.

[21] Appl. No.: **416,735**

[22] Filed: **Oct. 3, 1989**

[51] **Int. Cl.**[5] **G06F 15/21**

[52] **U.S. Cl.** **364/478**; 364/401; 364/464.02

[58] **Field of Search** ..... 364/478, 464.02, 464.03, 364/466, 401, 402, 200 MS File, 900 MS File; 235/375, 432; 270/54-58; 53/266 A, 504; 209/584, 900

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,639,873	1/1987	Baggarly et al.	364/478
4,797,832	1/1989	Axelrod et al.	364/478
4,800,504	1/1989	Durst, Jr. et al.	364/478
4,817,042	3/1989	Pintsov	364/478

Primary Examiner—Joseph Ruggiero

Attorney, Agent, or Firm—Peter Vrahotes; Melvin J. Scolnick; David E. Pitchenik

[57] **ABSTRACT**

A system and method is described for optimizing mail processing by applying a co-mail concept. A network is established involving a data center and plural publishers and inserters of advertising documents. Databases are established at the data center of profiles of the publishers and inserters and their current and prospective activities, and the information used by the data center to match publisher to inserters for particular jobs to reduce mail processing and distribution costs, or expedite mail distribution.

9 Claims, 6 Drawing Sheets

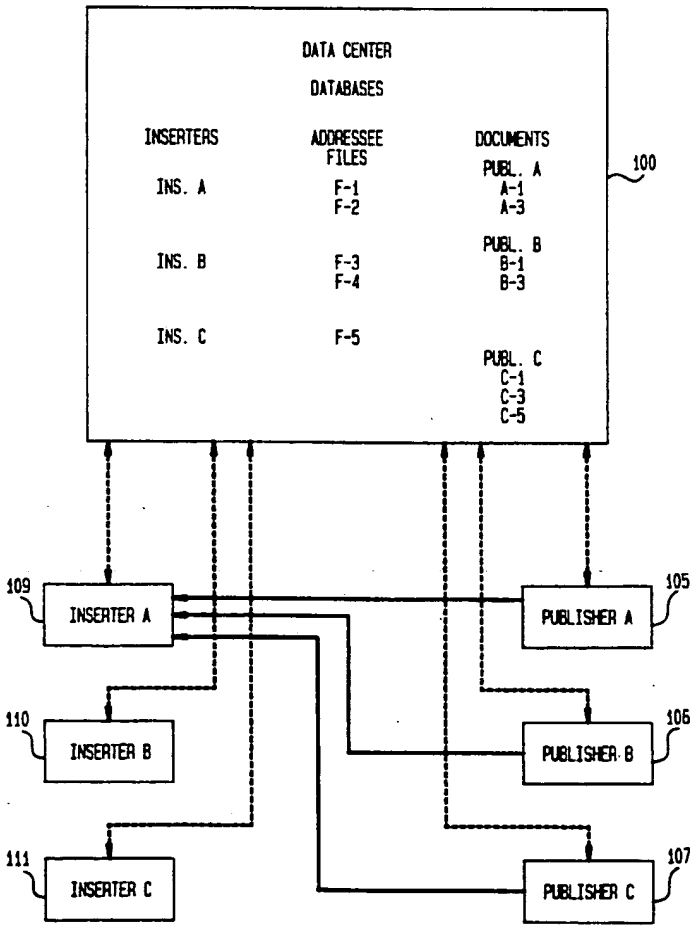


FIG. 1

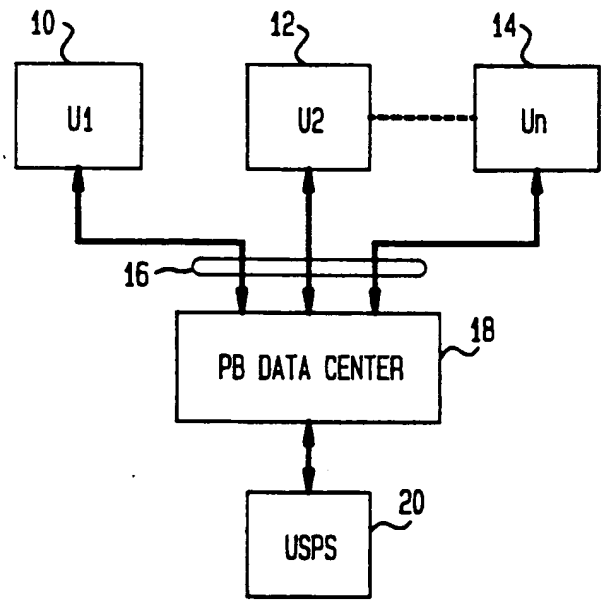
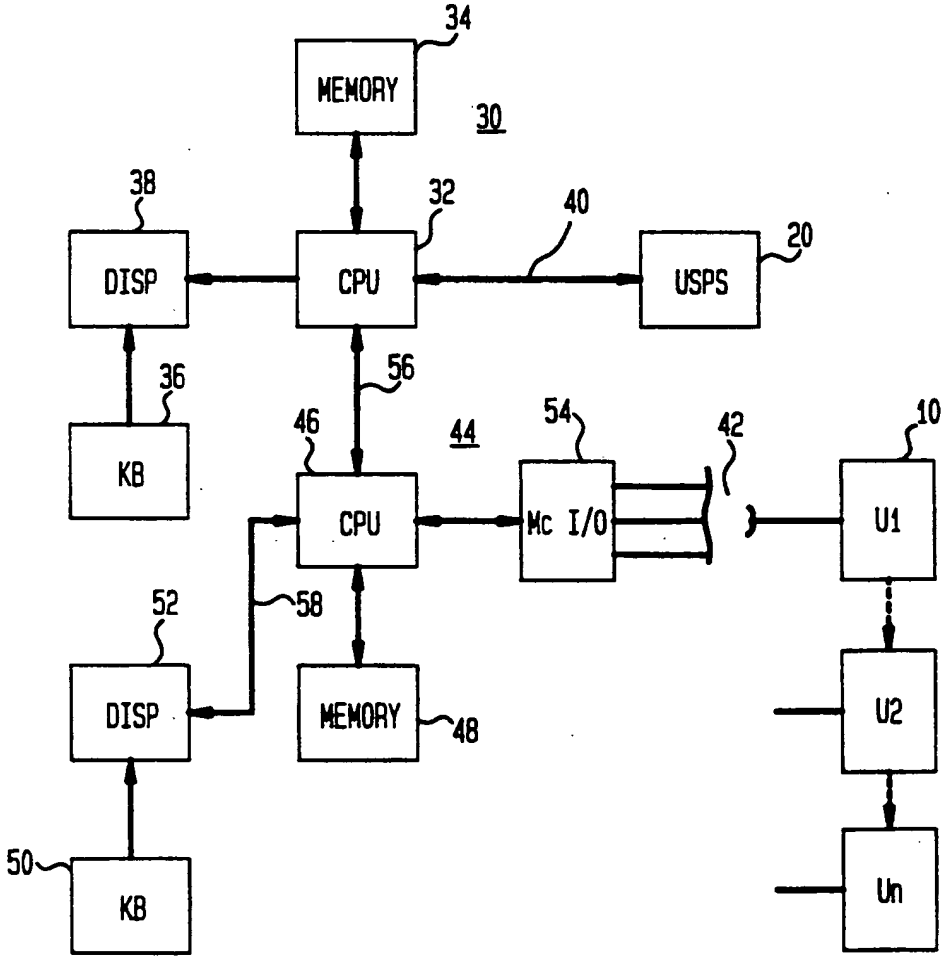


FIG. 2


34
MEMORY
30
38
32
40
DISP
CPU
USPS
20
36
56
46
44
54
42
10
KB
CPU
Mc I/O
U1
52
58
DISP
MEMORY
48
U2
50
KB
Un

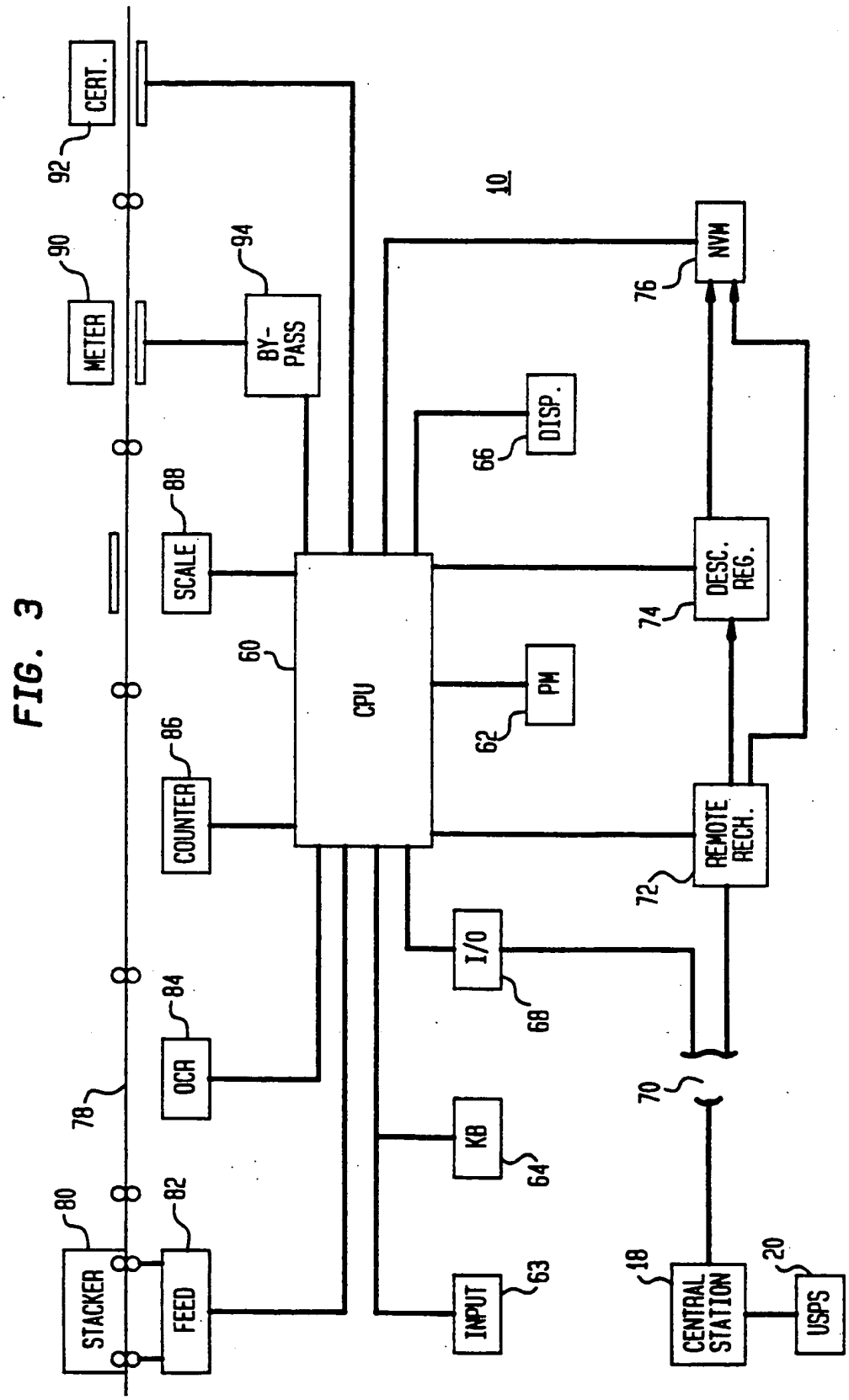

10
CERT.
92
METER
90
BY-PASS
94
SCALE
88
CPU
60
COUNTER
86
OCR
84
STACKER
80
FEED
82
78
DISP.
66
PM
62
KB
64
INPUT
63
I/O
68
NVM
76
DESC. REG.
74
REMOTE RECH.
72
70
CENTRAL STATION
18
USPS
20

FIG. 3

FIG. 4

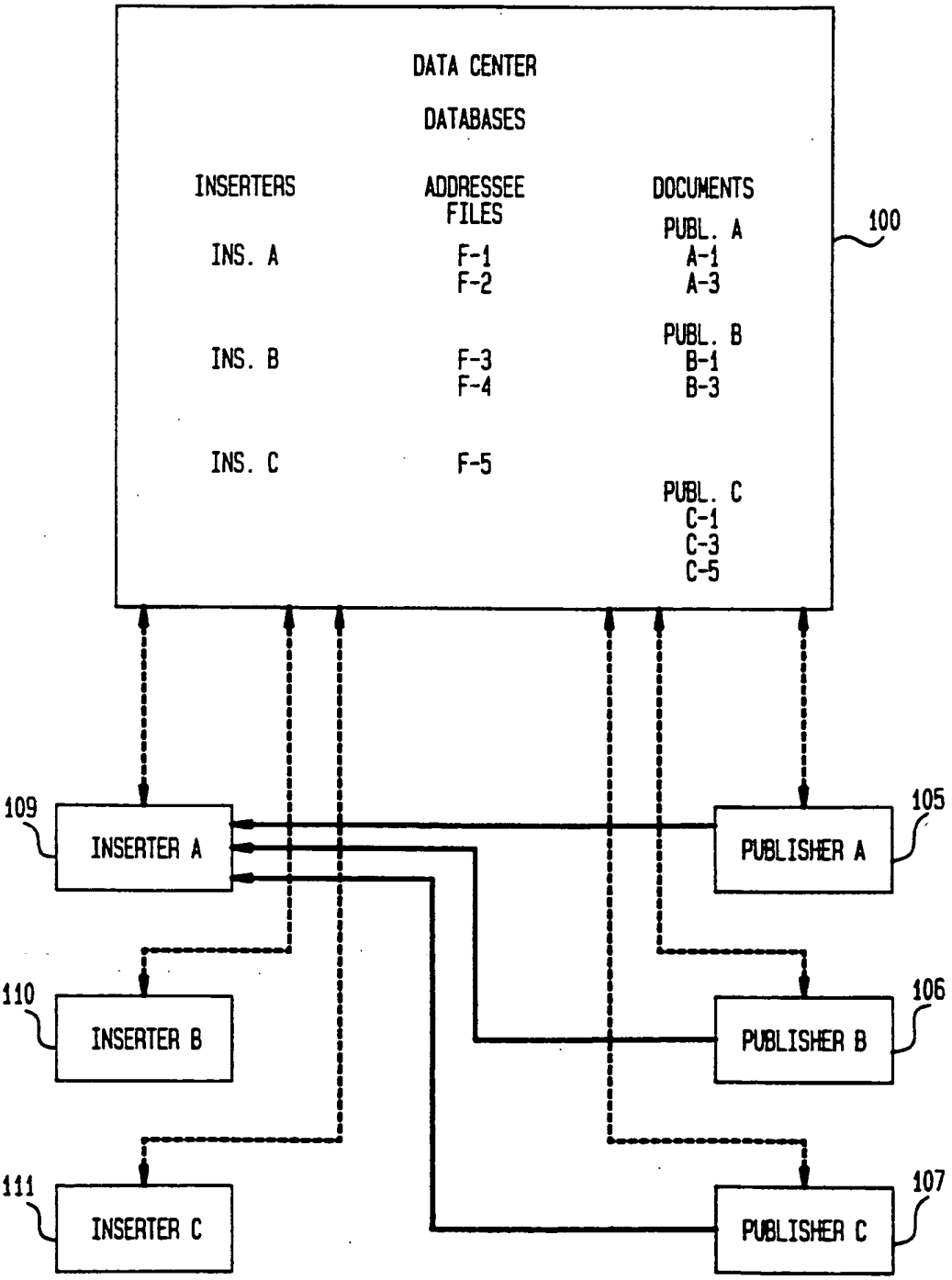

DATA CENTER
DATABASES
INSERTERS
ADDRESSEE FILES
DOCUMENTS
INS. A
F-1
F-2
PUBL. A
A-1
A-3
INS. B
F-3
F-4
PUBL. B
B-1
B-3
INS. C
F-5
PUBL. C
C-1
C-3
C-5
100
109
INSERTER A
105
PUBLISHER A
110
INSERTER B
106
PUBLISHER B
111
INSERTER C
107
PUBLISHER C

FIG. 5A

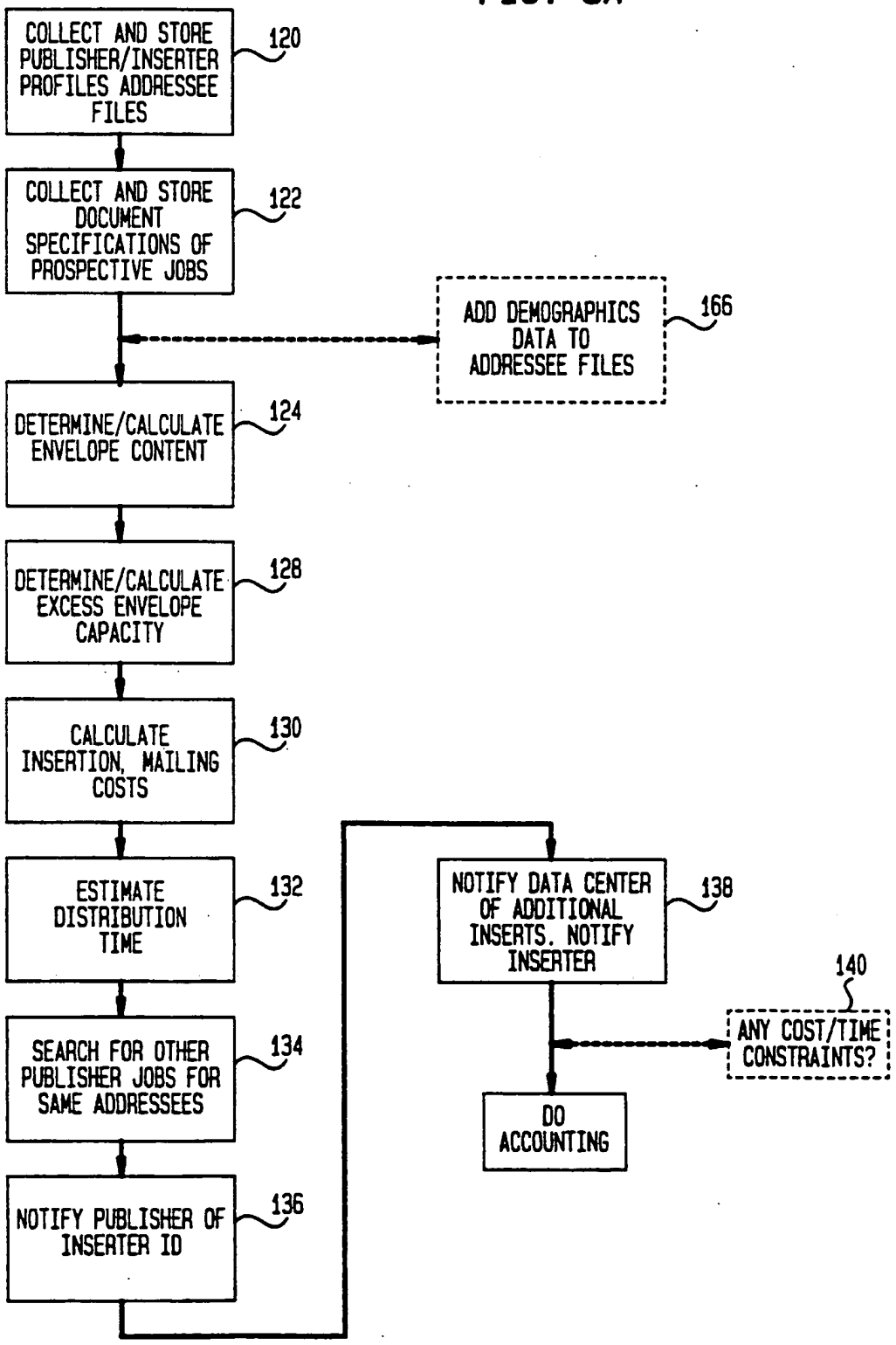
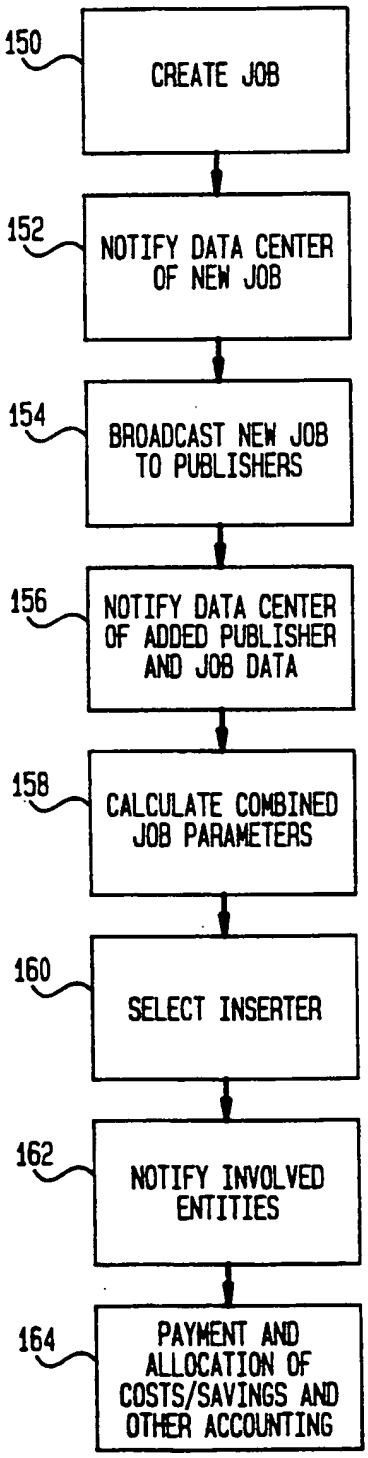

COLLECT AND STORE PUBLISHER/INSERTER PROFILES ADDRESSEE FILES
120
COLLECT AND STORE DOCUMENT SPECIFICATIONS OF PROSPECTIVE JOBS
122
ADD DEMOGRAPHICS DATA TO ADDRESSEE FILES
166
DETERMINE/CALCULATE ENVELOPE CONTENT
124
DETERMINE/CALCULATE EXCESS ENVELOPE CAPACITY
128
CALCULATE INSERTION, MAILING COSTS
130
ESTIMATE DISTRIBUTION TIME
132
SEARCH FOR OTHER PUBLISHER JOBS FOR SAME ADDRESSEES
134
NOTIFY PUBLISHER OF INSERTER ID
136
NOTIFY DATA CENTER OF ADDITIONAL INSERTS. NOTIFY INSERTER
138
140
ANY COST/TIME CONSTRAINTS?
DO ACCOUNTING

FIG. 5B


150
CREATE JOB
152
NOTIFY DATA CENTER OF NEW JOB
154
BROADCAST NEW JOB TO PUBLISHERS
156
NOTIFY DATA CENTER OF ADDED PUBLISHER AND JOB DATA
158
CALCULATE COMBINED JOB PARAMETERS
160
SELECT INSERTER
162
NOTIFY INVOLVED ENTITIES
164
PAYMENT AND ALLOCATION OF COSTS/SAVINGS AND OTHER ACCOUNTING

OPTIMIZING MAIL PROCESSING BY MATCHING PUBLISHER AND INSERTER ENTITIES

This invention relates to article processing and delivery, and specifically to improvements in mail processing systems to reduce costs and which will relieve official postal facilities of certain mail handling tasks.

BACKGROUND OF THE INVENTION

As discussed in commonly-assigned U.S. patent application Ser. No. 285,891, filed Dec. 16, 1988, U.S. Pat. No. 5,008,827 the contents of which are herein incorporated by reference, the United States currently has the world's largest postal system. The U.S. Post Offices currently handle in excess of 100 billion pieces of mail per year, about half the total volume handled throughout the world. The servicing of mail delivery involves three essential steps; collection, sorting and delivery. Collection takes place through a series of Post Offices spread throughout the United States. The United States has about 30,000 Post Offices that provide mail services in addition to 9,000 smaller postal centers which provide some type of mail service. Postal employees typically take letters and packages from mail box facilities to the nearest local office where it is accumulated for the sorting procedure. At the Post Office, postal clerks remove collected mail from sacks, bundle packages and segregate mail by size and class into separate categories. The mail travels by truck from local post offices to a central facility known as a sectional center. The United States has 264 sectional centers, some of which serve hundreds of local Post Offices. The sectional center processes nearly all the mail coming or going from its region. At the sectional center, high speed automated equipment sorts large volumes of mail. A computerized machine known as a zip mail translator sorts the postmarked letters according to their destination post office. Postal workers selectively activate the machine's keyboard to send each letter on a conveyor belt into one of many bins. Each bin holds mail for a different Post Office or postal region. Mail addressed to locations outside the regions served by the sectional center are transported by truck, airplane or train by various carriers via various routes to other sectional centers for further sorting. Finally, postal clerks hand sort mail for the area served by the local office into bundles for each delivery route.

Current estimates indicate that the foregoing processing tasks necessarily involve in excess of half a million employees. Cost of maintaining and supporting sorting services at the central post office facilities, even including large scale use of automated equipment, has become staggering. Projections of substantial increases in volumes of mail being transported through central facilities, even with the advent of private delivery, telecommunications services, facsimile services and the like indicate a rapid expansion will be required of such facilities. Since the Postal Service is a private corporation and is expected to become self-supporting, rapidly advancing postal rates place greater and greater burdens on both users and the Postal Service in order to support such volumes of mail. In recognition of capabilities of certain high volume users to provide services to central postal facilities, which services may improve efficiency and reduce the amount of processing time required by the central serving facilities, the U.S. Postal Service

offers substantial reductions in rates or discounts, provided that a user comply with certain requirements which will allow the U.S. Postal Service to take advantage of certain user-provided facilities to reduce its own work load. The concept of work sharing, wherein a user provides certain of the processing activities prior to delivering the mail to the postal system, has been proposed in the copending application and is therefore a positive innovation in the field of mail processing which may have a substantial impact in the future implementation of mail services.

Under present systems in use, entities involved in creating batch mailings unwittingly contribute to the increasing workload of the Postal System. In mass mailings especially of advertisements (ads), it is common to print the hardcopy ads at a printer local to the ad publisher, and package ad mail at one or a few professional mailer entities. If, for example, the ads are destined for distribution over a wide geographical area, but the mailings are generated at one or only a few locations, it means that the Postal Service assumes the major burden of processing and delivery to the addressees.

Another factor contributing to inefficient distribution is envelope capacity. Frequently, ads are generated at several different locations, separately enveloped or packaged (meaning machine stuffed in mailing envelopes or packages), addressed, and then dropped off at a local Post Office. Often these separately enveloped ads are intended for the same addressees. So, the Postal Service ends up delivering separate mailings to the same addressee, resulting in multiple processing of documents delivered over the exact same route.

DESCRIPTION OF RELATED APPLICATIONS

Systems relating to work share features are discussed in copending commonly-assigned applications Ser. Nos. 234,977, U.S. Pat. No. 5,005,124, U.S. Pat. No. 5,008,827, and Ser. No. 285,891, filed Aug. 23, 1988 and Dec. 16, 1988, respectively, the contents of which are hereby incorporated by reference. These applications relate to the concept of using certain limited user provided services but does not encompass the full range of work sharing and data sharing services and concepts presented herein.

Commonly-assigned U.S. Pat. No. 4,713,761 describes a system for determining and accounting for the costs of shipping goods, which focuses mainly on a centralized accounting facility for multiple shippers and carriers.

Commonly-assigned U.S. application Ser. No. 285,486, filed Dec. 16, 1988 describes and claims another data sharing system via a centralized database to improve user performance, the contents of which are incorporated herein by reference.

Concurrently-filed, commonly-assigned U.S. application, Ser. No. 416,732 describes and claims other aspects of systems for improving delivery efficiency.

Commonly-assigned U.S. Pat. Nos. 4,800,504; 4,800,505; and 4,800,506 describe computerized systems for preparing stationery items and their subsequent processing into mailpieces.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a system and method that enables a more efficient and effective use of the facilities of the post office.

A further object of the invention is a system and method for optimizing delivery of bulk mail from mailers to a post office.

Another object of the invention is to reduce costs for entities involved in generating batch mailings.

Still another object of the invention is expedited delivery of mass mailings to their respective destinations.

The present invention extends the work-sharing concept by what can be termed the concept of co-mail, meaning, cooperation of the many entities involved in the creation of batch mailings in order to minimize costs or expedite mailings or reduce the Postal Service workload. This is accomplished by building on the data communications network described in several of the above-related applications to include on the existing network or as part of an independent network various entities involved in the creation or generation of mass or batch mailings.

In accordance with one aspect of this invention, a data center is provided in a data communications network with a plurality of entities involved in the business of generating ads or cyclical mailings for certain kinds of addressees. In addition, a plurality of inserters are also linked to the network. By "inserter" is meant those business entities that possess modern inserter equipment, typically under computer control, that has the capability to selectively collate ad sheets from stations distributed along a conveyer, and then insert or stuff the collation, folded or unfolded, into an envelope, which, typically, is then sealed. In addition, such equipment often include scales, printers for permit mailings, or postage meters. The envelope content is frequently determined by the particular addressee list for which the collation is destined. Knowledge of the existence of a mass mailing of mail pieces intended for particular addressees is a valuable commodity, because there may be other entities in the mailing business who also have or can generate additional items that can be included in those same envelopes at little additional cost due to the sophisticated capabilities of intelligent inserter equipment. In other words, if these additional items were simply added to the insertion line, and the computer accordingly adjusted, they could very easily be added to the very same envelopes.

What the system of the invention provides is an online mechanism for informing entities using inserters of inserter activities and the nature of those activities that can profitably be availed of by those entities. It does this by establishing a database containing information concerning the business operations of the inserter entities, including location, the inserter equipment available, and addressee files available, that is, lists of addressees owned or accessible to the inserter entity and to whom it frequently directs mail articles. The database could also include what other mail processing equipment is available to the inserter. In addition, the data center establishes a database of inserter entity users, such as publishers, which could include their location, kinds of ads generated for what kinds of customers to what particular sets of addressees. As a result of processing of the data, the system will allow inserters to, in effect, broadcast through the network current jobs, addressee profiles for whom intended, and excess envelope capacity available for third party advertising. To perform this function, the data center does not require the actual addressee file (listing of addressees by name and address), but only the number of addressees and their profile, such as, location, income level, housing status, etc.

In accordance with another aspect of this invention, the data center maintains regular communication with both the publisher and inserter entities via the network. This allows the data center to maintain up-to-date information not only on current jobs in execution but also on prospective jobs to be carried out and backlog of jobs to be executed. As much of this information may be business proprietary, the data center, being independent of both the publishers and the inserters, can by known security devices easily maintain such information confidential. Armed with these information databases, it is a relatively simple matter using conventional database processing to process the databases to select or match one or more inserter entities to execute a particular job of one or more publishers. Typical matching criteria are: job completion time, job costs, mailing and distribution costs, and mail delivery times. An important criteria would be the mailing profiles of the ultimate addressees who will receive the ads.

In accordance with another aspect of the invention, the data center would also maintain a database of the job content of now executing or soon to be executing prospective jobs. By "content" is meant the size and weight of the printed ads, for what kind of addressees are the ads intended, envelope requirements, etc. This would allow the data center to inform publishers of inserters who are preparing job mailings that would have additional capacity in their mail pieces to add additional material at no additional postage or at minimum additional postage. The publisher could then send its mailings to that inserter for inclusion in the envelopes. Or, it may be that an inserter is executing a job for a first publisher for distribution to a particular mailing list. A second publisher who has ads intended for the same addressees on the mailing list would save expenses by adding its ads to the same envelopes, with the first publisher sharing in the savings. Not only do both publishers benefit, but so does the Postal Service, which now has only one mail piece instead of two to be delivered to each addressee.

Many other situations can arise where matching of publisher jobs to printer jobs can profit both sets of entities.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing brief description and summary of the invention will become more apparent from the following more detailed description of the invention, accompanied by the attached drawings, wherein:

FIG. 1 is a functional block diagram illustrating the relationship between a data center, the U.S. Postal Service, and local users;

FIG. 2 is a more detailed block diagram illustrating the system of FIG. 1;

FIG. 3 is a more detailed block diagram illustrating a typical user or mailer station;

FIG. 4 is a schematic diagram illustrating operation of one form of system according to the invention;

FIGS. 5A and 5B are flow charts depicting several typical processing operations of a system according to the invention.

To best understand the system of the invention, it would be helpful to summarize the network description given in the referenced copending applications, because the same kind of network with only minor changes can be employed in the present invention.

SUMMARY OF PRIOR APPLICATION DISCLOSURES INCORPORATED HEREIN

With reference to FIG. 1, a plurality of user stations designated as U_1, $U_2 \ldots U_n$, and identified as **10, 12** and **14** are shown. It will be understood that multiple user stations are possible in excess of the three shown, and that these are shown by way of example only. These stations are coupled by means of an interconnection data communication network, illustrated generally as **16**, to the data center **18**, which in turn may be appropriately coupled by means of a secure data line or the like to the U.S. Postal Service **20**. The data center is a facility run by a commercial operation, such as Pitney Bowes, Inc., the assignee of the present invention Each of the blocks **10, 12, 14, 18** and **20** contemplate the use of data processing components, each appropriately interlinked by means of high speed telecommunication links or the like for the purposes of exchanging information. It is also contemplated within the scope of the invention that the U.S. Postal Service will maintain an appropriate computer facility, not specifically described herein, which will possess the capability of uploading and downloading specific pieces of information upon request by the data center, and relating to appropriate postal rules and regulations which will effect the use of certain discounts in mailing postal rates, as well as other factors necessary for the concept of shared work services which will be certified by each of the individual user stations in order to qualify for additional discounts when mail is received in the U.S. Postal Service facilities. The specific pieces of information will also be information relating to Postal Service planning and operations, as well as carriers that may be available for use and routing. The communication link is also contemplated as a two-way link between units **18** and **20**, wherein the U.S. Postal Service can have, if desired, the capability of monitoring specific operations within the data center in order to ensure the data center is operational in accordance with rules and requirements which may be imposed by the Postal Service from time to time. The monitoring operation can be periodic unscheduled communication link examinations of certain storage areas of accessed memory locations for confirming proper operations. Of course, visual on-site inspections and examinations may also be made.

With reference to FIG. 2, a more detailed functional component relationship of the system of FIG. 1 is illustrated. Thus, the data center includes a first data channel **30** which includes a CPU **32** having a program memory **34**, a keyboard **36** and an appropriate display **38** coupled thereto. Data communication link **40** interconnects CPU **32** to the U.S. Postal Service **20**. The user units **10** are coupled via appropriate telecommunication data links **42** to a second data channel **44** which includes a CPU **46**, a program memory **48**, a keyboard **50** and an appropriate display **52**. The CPU is coupled to the data links **42** by means of a multi-channel IO device **54** capable of high speed data communication.

In operation, two-way communication between the data channel **30** and the U.S. Postal Service **20** provide a continuous interchange of information regarding updates of U.S. Postal Service rules and regulations covering rate reductions and as may be required for the continuous certification use by the local users **10**. In addition, the data channel **30** may also be manually interrogated by means of keyboard **36** for inquiring of the Postal Service for specific information which may be employed with regard to compliance with certifications, the answering of specific data questions, or other uses requiring specific interrogation by means of the central station to the U.S. Postal Service. Since the U.S. Postal Service link is a two-way communication over channel line **40**, it is possible through this link for the U.S. Postal Service to interrogate and monitor the operation of the first data channel **30** and the second data channel **44**, for compliance with quality control and other security compliances which may be required by the U.S. Postal Service.

Turning now to the second data channel **44**, high speed continuous two-way communications with respect to continuous update of U.S. Postal Service requirements for certification, servicing and diagnostics, training, and other information interchange, as described in application Ser. No. 285,891, are effected by means of the CPU **46** operating through the high speed data channel **54** interfaced along with communication lines **42** to the multiple user network U1, U2 . . . Un. Operating under control of the program memory **48**, the CPU **44** is contemplated as a high speed multiple processing information apparatus of conventional design such an IBM **3083** or a CD VAX unit which may handle multiple requests from any one or more of the users simultaneously through the multiple channel I/O device. Keyboard **50** and display **52** may be utilized for manual information interchange between any of the local users and the information operator. Although only a single keyboard display unit is illustrated, it will be understood that dataline **58** is a schematic representation of the existence of a multiple number of display keyboard combinations evidencing the use at the central station of a plurality of key operators available to answer user questions upon interrogation.

Referring to FIG. 3, a function block diagram illustrating the interrelationship of typical components within each individual user station is illustrated. The central user station operates under the control of a CPU **60**, which includes a RAM memory and appropriate control registers. Coupled to the CPU **60** is a program memory **62** which defines the essential function of the system, including updating instructions and rates used in the local user units, diagnostic monitoring, a two-way communication link establishing a tracking facility utilizing, for example, the encryptic key represented by the certification, and interface capabilities with respect to the central station for the downloading of training information which enables operators at local stations to understand and comply with specific requirements imposed by the U.S. Postal Service with respect to the certification process. These capabilities are described in more detail in the related applications Ser. Nos. 285,891 and 285,486. Coupled to the CPU **60** are a keyboard **64** for the manual entry of data requests and other information into the CPU, display unit **66** and an I/O channel **68** coupled by means of a data link **70** to the central station **18**. Accounting information and funding to the local user **10** is effected through the data link **70** from the central station **18** to the remote recharger mechanism **72**, operating in accordance with conventional recharging techniques, such as disclosed in U.S. Pat. No. 4,797,045, assigned to the assignee of the present application. Remote recharging unit **72** charges a descending register **74**, in conventional manner, which keeps track of descending balances charged from time to time in accordance with mail requirements. Non-volatile memory unit **76** is employed to maintain security of informa-

tion during periods of time when local user unit 10 is not operating. Non-volatile memory 76 receives descending register balances as part of a shutdown routine, along with other security data which may be applied from the active memory of the CPU 60. Remote recharging operations are carried on in remote recharging circuit 72 via the data link 70 to the central station 18, and operates independently of the OFF/ON status of the local user unit 10 for monitoring purposes. Thus, even if the local unit 10 is turned off, central station 18 through data link 70 may inquire through the remote recharging unit of the status of certain pieces of information which are maintained either in the RAM memory portion of the CPU 60 during on times of the unit 10 or in the non-volatile memory 76 during inactive status periods. Another accounting system is described in commonly-assigned U.S. Pat. No. 4,796,193.

A concept of work sharings entail the performance of certain postal service functions by the user in a secure manner so as to enable the user to apply not only postage but to also apply certification, as an imprint on the mail piece, which will be accepted by the postal service that the services certified were in fact performed by the user and thus enable the user to be entitled to further mail rate reductions. Communication in contrast may also be by means of a code or other form with the relevant information transmitted in encrypted format. The information may be scanned and used to automatically set the postal equipment at the user site to proper settings, both for postage and for usage scheduling, without direct user intervention, thus enhancing security and efficiency.

Certification information is provided to the CPU through a plurality of inputs along a mail path designated as 78. Mail piece documents which are stacked in appropriate feeder-stacker unit 80 are, under control of CPU 60 through feeder-unit 82, driven along the mail path 78, past OCR unit 84 where printed material on the mail piece is read, past counter station 86 where individual pieces are counted, to the scale unit 88 where the mail piece is weighed, and thence to a metering station 90 for application of appropriate postage and finally to a certification station 92 where appropriate certification stamps may be placed on the mail piece to indicate compliance of the mail piece with all the criteria that have been set under work sharing requirements required under relevant U.S. Postal Service regulations. Since the unit may be capable of handling prefranked mail, a meter bypass network 94 operating under control of the CPU, provides for bypassing of the mail piece of the metering station 90 without the necessary application of additional postage. Problems encountered in short-weight mail may be adjusted by appropriate decrement of the descending register balance in descending register 74 under program control through CPU 60, based upon differences detected by the computer between applicable postage rate requirements and the actual mail run being passed through the user station 10. An example of short-weight mail is disclosed in copending application Ser. No. 285,146, U.S. Pat. No. 5,019,991 filed concurrently herewith, and assigned to the assignee of the present application.

The data exchange between the central station and the user station give rise to further novel and unique processing capabilities in accordance with copending application Ser. No. 285,486. The user equipment can be periodically interrogated as to the customer usage patterns, mail runs, addressees, zip codes, rate breaks, etc. Based on this usage information, information can be down loaded to the user. The down loaded information includes advisory procedures, such as the fact that postal charges for first class mail will increase shortly and that any mail run should be pushed out early, the fact that the user has the ability to avail itself of zip code breaks and other useful information as to mailing habits. In mail addressing, the data base incorporated by the central station also can benefit the user by expansion of the user equipment far beyond its limited nature by taking advantage of the relatively larger data base facilities available at the data center. One particularly advantageous use of the communication capability is the uploading of user mailing lists to a data center. At the data center, the information can be processed relative to the data center data base to sanitize the mailing address list. This includes noting improper zip codes, changes in carrier routes, changes in addresses when people move, etc. The information is thereafter down loaded to the user mailing equipment at the user location to allow the generation of sanitized effective mailing lists. This enables the user to have a higher "hit" rate of properly addressed and properly delivered mail. Moreover, the system provides a tremendous advantage to the U.S.P.S. Billions of dollars are spent in redelivering mail which is improperly addressed. Mail can be improperly addressed for a number of reasons including the fact that an individual has moved and the new address data has not been properly entered into the system. By providing this service to, say, the top 500 mailers in the country, which constitutes in the are of 80% of the mail in the country (3rd class), a tremendous amount of expense would be saved by the Postal Service. Where the mailing lists are small, the businesses can upload their mailing lists to the data center. The uploaded mailing address are sanitized and processed, and in this instance, the center can provide the actual mailing service for the small business by providing the printing of the necessary inserts, inserting the material into the envelopes and addressing and distributing them. Alternatively, the center can download the address list to the mail business in the appropriate format. It can also be presorted by order with an indication as to the appropriate postage amount. The postage chargeable when the data center provides the mailing service can be directly charged to a user account. Additionally, demographic analysis can be done on the mailing list to identify additional customers who would fall into similar categories and are not yet being serviced by the small business. Moreover, by demographics analysis, other business opportunities for customers of the type serviced by the small business can be identified and provided to the user. The system would include sufficient security to ensure that the mailing list data for the customer is not improperly utilized. This has been described in more detail and claimed in a copending application Ser. No. 285,145, filed Dec. 16, 1988, and concurrently filed application Ser. No. 416,732 which describes and claims extensions of these concepts.

DETAILED DISCLOSURE OF PREFERRED EMBODIMENTS OF PRESENT INVENTION

For the network described above to be operable for carrying out the present invention, additional entities have to be added. These entities would have on their premises the same kinds of computer equipments described for the users in the earlier network, and other equipment, all of which is conventional and commer-

cially available, for carrying out the functions of that particular entity. Thus, a publisher entity would have the usual resources currently possessed by publishers, including personnel and equipment for creating ad cop and job specifications for that copy. Publishers would also have available a means for recording and storing in their computer the physical parameters and specifications of each ad copy. Moreover, if the publisher also were a mailer, it would have as resources the same mail processing equipment above described for the user stations In addition, the publisher could have the capability to create E-Mail versions of its ad copy, which can then be transmitted over the network or via telephone lines to other entities on the network.

Printer entities would have as resources the usual equipment to convert ad copy into hard copies of the ads, in the final form to be distributed to addressees. Printers sometimes have mail processing capabilities also. That is, not only do they print the hard copies of the ads, they also have the equipment to stuff or insert them into envelopes, seal and stamp them. Printers sometimes have addressing capability. That is, they have addressee files and can use those addressee files to address envelopes for the publisher and then to package those envelopes for delivery to the Postal Service.

Inserter entities, nowadays, have what is known as intelligent inserters which, as previously described, under computer control, provides a series of stations holding multiple copies of a particular insert, and whereby collations are advanced through the machine past each of the stations which are programmed to add inserts to particular collations. At the end of the machine, the collations are stuffed into envelopes which are then sealed and subsequently processed for franking or other postage accounting. An example of one such machine is described in commonly-assigned U.S. Pat. No. 5,003,485, filed Dec. 30, 1988. Commonly-assigned U.S. Pat. No. 4,817,042 also describes a computerized inserter machine for optimizing the final weight of a mailpiece. Inserter entities frequently have equipment for addressing envelopes in accordance with addressee files of particular profiles. In addition, the inserter entities would have the same kind of computer equipment as described for the user station in FIG. 3, that would enable the inserter, as any other entity on the network, to be in regular communication with the data center.

In the system of the invention, the data center maintains a database of publishers and their profiles, and a database of inserters and their profiles. Printers can also be added if desired. Moreover, each of these user entities on the network are continuously uploading to the data center information concerning their current activities and planned future activities. The data center processes this information seeking profitable matches between the various entities. Examples of suitable matches will be given below, but first, reference is made to FIG. 4 which provides a diagram illustrating the relationships among several of the entities.

In FIG. 4, the data center is represented by block 100. It maintains databases of publishers on the network, represented by blocks 105, 106 and 107, and databases of inserters on the network, represented by blocks 109, 110, and 111. Each of the latter is in continuous communication with the data center, represented by the double-arrow-ended dashed lines. As a result of these communications, the database has stored the following information, as an example only. Publisher A has ads A-1, and A-3 to be distributed, respectively, to addressees in files F-1, and F-3. Publishers B and C also have ads as similarly labelled in block 100 to be sent to the addressee files indicated by the suffix. Note that all three publishers, operating independently, have created ads intended for addressees with the profiles of the F-1 file.

The inserters can be located in different geographic regions. Their capabilities to insert the ads of the publishers, whose specifications may be maintained at the data center, is readily determined by the data center. Due to continuous on-line communications, the data center can choose an inserter for a particular job and take into account not only equipment and other resources available to each inserter, but also current workloads and the ability to complete the job within the time frame desired by the publisher. Another important factor easily included in the data processing is the locations of the addressees relative to the location of the inserters For instance, suppose that addressee file F-1 is composed of addressees located in a certain region of the country. These addressees are to receive, say, a monthly statement from a utility, credit car company, or bank, and/or additional messages or ads. Distribution of this information or ads to such addressees would be expedited by choosing an inserter located in that region if the selected inserter had the resources to make up the mailings to those addressees and deliver to the local Post Office. So, already valuable benefits accrue both to the publishers and inserters as a result of the operations of the data center, as an independent entity, in maintaining these databases of publisher and inserter activities.

But, other even more valuable benefits can be obtained from the system of the invention. Suppose Inserter A has been hired to insert ad A-1 from Publisher A into envelopes, seal, address to addressees on file F-1 and deliver to the local Post Office. The data center has detected from its databases that Publishers B and C also have ads intended for the same addressees in addressee file F-1. Accordingly, Publishers B and C are notified to forward their copy to Inserter A to include in the same envelopes that will contain Publisher A's copy. All members of the system gain from this example of co-mail. Postage costs are minimized, as it may not cost too much more to add Publisher B's and C's documents to the common envelope. Moreover, distribution to the addressees due to use of local Post Offices is expedited. The Postal Service benefits, as it now has only to deliver one letter instead of three.

There are many other ways in which the co-mail principle can be applied to benefit the network entities. Suppose that Inserter B has received an order from Publisher B to print and mail documents B-3 to addressees listed in file F-3. The data center is informed. The data center, storing profiles of the publisher's activities, knows that Publisher C also frequently creates copy for addressee file F-3. The data center can then advise Publisher C that Inserter B is inserting documents for distribution to addressees in file F-3, that the weight of the documents from Publisher B is only one-half of the weight allowed for the postage to be supplied, and thus envelope capacity is available if copy for the same set of addressees exist and is forwarded to Inserter B in time. They then can be added to the existing envelopes for no additional postage. Thus, the postage costs for Publishers B and C are halved in this example of co-mail.

The previous example is just one instance of how the data center, knowing from its up-to-date databases of current and prospective inserter jobs, can easily process its stored information to find publishers that can utilize

excess envelope capacity developed by the inserters. This not only saves postage costs in the distribution of the ads, but also will allow speedier distribution of the ads to the intended addressees. Moreover, the data center can contribute additional value to the networked entities by participating in the delivery of the ads or physical inserts from the publishers to the inserters. So, for example, if the ads are created on a desktop publishing workstation, they can be transmitted electronically over the network to the data center, and then from the data center to the selected inserter. If in physical form, the ads can be delivered to personnel at the data center who would then arrange for delivery to the inserter.

Another valuable contribution of the data center is accounting for the networked entities. The data center can maintain accounts for the networked entities, and credit inserter's accounts for payments by publishers and similarly debit the publisher's account. In the system of FIG. 2, wherein the Postal Service can be linked to the network, payments to the Postal Services for postage can be accomplished by the data center on behalf of inserter mailers. Also the data center as a third party can fairly allocate any postage savings among the operating entities. Commonly-assigned U.S. Pat. No. 4,837,701 also describes an accounting system employing a central station, whose principles and configuration can also be used herein for payment accounting.

There are still other ways in which a data center, linked in a network with and possessing information concerning activities of mail publishers and mail inserters, can benefit the entities involved. Another important way is by providing value-adding-information to users with limited resources of their own. For instance, a data center subsidized by many users can compile and process much valuable information for entities on the network. Thus, there are organizations that create demographics information on individuals and families throughout the country. The cost of acquiring such information and customizing it for particular publishers or its customers can be prohibitive for one entity, but affordable when performed by the data center for many entities. The data center can use this demographics information to update addressee files of inserters and publishers, to add or delete addressees, or to form smaller sets of addressees for more focussed merchandising.

This is of benefit to the networked entities because the value of addressee files used by inserters to publishers depends upon its accuracy and its granularity, i.e., the finer grained the listings, meaning, the finer subdivided the addressee sets can become, the more effective will be target marketing and pinpointing of potential customers. Such demographics information is available from several institutions. For instance, LPC, a Pitney Bowes company, provides a Geographic Data File which is a directory of geographic locations cross-referenced by zip code. This information is readily appended via known software to an address in a database file of addressees. Similar software is available for adding demographics data broken down into over 300 variables for many target marketing applications.

FIGS. 5A and 5B are flow charts depicting typical sequences in the operation of the system of the invention. In block 120 of FIG. 5A, the data center uploads from each of the entities on the network the necessary information to establish profiles of the business activities of the publishers and inserters, which profiles would assist matching publisher jobs to inserter jobs or capabilities. In addition, the data center would store addressee files available to inserters or mailers or used by publishers for their ads:

In block 122, the inserters upload to the data center data representative of current and prospective jobs, which would include the type, weight and size of ads, volume of ads, envelope types to be used, and for which set of addressees intended.

The data center would then process the information it has stored, using known database processing techniques, to determine by calculations the ad content of the envelope block 124, whether excess envelope capacity will be available at block 128, insertion and mailing costs at block 130, and estimated distribution time at block 132. Then, at block 134, the data center computer, using known database processing techniques, would scan through the stored data seeking other publisher jobs intended for the same set or subset of addressees and that would fit the available excess envelope capacity. At block 136, the identified publishers are notified of the pending inserter job. Assuming a publisher decides to take advantage of this opportunity, at block 138 the data center is notified, which in turn notifies the inserter to put the job on hold pending receipt of the additional inserts from the second publisher to be added to the collation to be stuffed into the envelopes to be addressed to particular addressees. The addressing can take place at the inserter or at a third party mailer or at a publisher. The publisher selection, if desired, could take into account any time and cost constraints imposed by the publisher, shown at block 140 in dashed lines. If more than one publisher decides to use the excess envelope capacity then the selection could be based on such factors as geographic location of the second publisher relative to the inserter involved. If necessary, the data center can become involved in transmitting the new inserts, in physical or electronic form, to the involved inserter, and also in the accounting of the transaction shown at the block 142.

FIG. 5B shows a variation. In this sequence, a publisher decides to mail certain inserts to a set of addressees, at block 150. The publisher notifies the data center of this prospective job and its specifications at block 152. At block 154, the data center broadcasts this prospective job to other publishers who may desire to share the available envelope space to reduce costs. Assuming a second publisher decides to accept this opportunity, the data center is notified at block 156, together with specifications on these new inserts. At block 158, the data center calculates the job parameters, determines from its database of inserter profiles which inserter is best suited, at 160, to execute the job, based on time and resource availability, and notifies, at 162, the entities now partaking in this co-mail venture. This broadcasting sequence, originating with a publisher, can also originate with an inserter, for example, during a slack period and seeking additional work, perhaps at a discount. The inserter if a mailer may also have proprietary addressee files, whose availability can be broadcast to the publishers for their use. The previous sequence of steps involving calculating envelope capacity and seeking additional inserts to fill available envelope space can also be followed in this example. The data center can also scan through its stored listing of documents or jobs and prospective jobs to determine whether any would qualify for incorporation in those envelopes. In other words, if there are any other mailings available destined for the same set or for a subset of

the addressees, the publishers of those jobs would be notified that a job is being executed destined for a set of addressees to whom the publisher also intends to send documents; hence considerable savings in time and money are available by taking advantage of unfilled envelopes destined for those addressees.

Finally, in block **164**, the data center, if desired, can arrange for payment to the inserters upon completion of the jobs, and debiting the account of the involved publisher, if alone, or allocating the cost thereof among the cooperating publishers in accordance with any formula agreed by the publishers.

As shown in dashed lines in FIG. 5A, at block **166**, if desired, demographics data can be added to the stored addressee files for targeting refined subsets of addressees. These augmented files can be returned via the network to the owner, inserter or printer, or stored at the data center.

As noted, by relatively simple, straightforward programming, the matching of publishers to inserters can be readily optimized in accordance with the needs and resources of the entities involved to the benefit of all members on the network. The public benefits because reduced distribution costs will be reflected in reduced merchandise prices, and because, for the same and in most instances improved targeting of prospective purchasers and reduced mailings, the involved Postal Service operations are also reduced.

The system and methods described herein can be used alone, or in combination with many of the systems described in the referenced copending applications. For example, the concurrently-filed application Ser. No. 416,732, describes a similar network in which the entities matched are publishers and printers. That system is readily adapted to be consolidated with the present system by adding printers to the network as there described. Also, as another example, the referenced copending applications Ser. Nos. 285,145 and 285,486 relate, respectively, to networks which allow updating of customer lists by a data center, and to networks in which a data center creates databases of mailer activities for selective accessing by mailers to improve their operations. Those systems are readily integrated with the system of the present invention, since the same equipment can be used, the major difference being the information stored at the data center and the data processing criteria. The same applies to the certification system described in application Ser. No. 285,891, U.S. Pat. No. 5,008,827.

It will be evident to those skilled in the art that, from the network descriptions given herein and in the referenced Applications, it is a relatively straightforward development to construct such a network and establish the data communications links that will allow the data center to perform the functions described herein. Also, those skilled in the art will recognize that creation of the databases described, and programming of the data center computer to process the stored information to make the desired matching of publisher to inserter is also a straightforward project development using programming and development tools available from many vendors. Any known, suitable, matching or selection algorithm can be employed, in addition to the preferred ones described above. The processing time is not critical, and thus, while desirable, it is not critical to the invention that the databases are built and processed rapidly or that access time to the network is short.

While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled in this art may be made therein without departing from the spirit of the invention, and the invention as set forth in the appended claims is thus not to be limited to the precise details of construction set forth above as such variations and modifications are intended to be included within the scope of the appended claims.

While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled in this art may be made therein without departing from the spirit of the invention, and the invention as set forth in the appended claims is thus not to be limited to the precise details of construction set forth above as such variations and modifications are intended to be included within the scope of the appended claims.

What is claimed is:

1. Apparatus for performing mail processing functions for postal system users, said apparatus comprising:

a data center having a data processor and means for receiving via a data communications link and storing data representative of:

(a) activity profiles of plural inserter entities having intelligent inserter equipment,

(b) current or planned mail documents to be generated by a plurality of publishers;

means for processing said data to determine which of said mail documents should be forwarded to which of said inserter entities to reduce costs or expedite distributing of said documents to addressees.

2. Apparatus as claimed in claim **1**, wherein the activity profiles include addressee files available to the inserter entities and available envelope capacity for determined addressees.

3. Apparatus as claimed in claim **2**, further comprising means at the data center for storing demographics data on prospective addressees and for transmitting said data to an inserter entity.

4. A system for performing mail processing functions, comprising:

(a) a data center having a data processor and storage means,

(b) a plurality of entities having intelligent inserters for inserting documents into mailing envelopes,

(c) a plurality of entities performing publishing functions and capable of generating documents or document specifications for documents for distribution to a set of addressees,

(d) a data communication network interconnecting the data center with the inserter and publishing entities whereby said inserter entities can transmit to the data center data representing their inserter activities including mailing envelope capacities and addressee files, said data center data processor storing said data transmissions in said storage means,

(e) said data center processing said stored data for selecting any of the publishing entities on the network that can utilize particular inserter envelope capacities or addressee files for its documents for distribution to said addressees,

(f) means for transmitting to said selected publishing entities the identity of said particular inserters and its available envelope capacities or addressee files.

5. The system of claim **4**, further comprising means at said data center for supplying to particular inserter

entities via the network demographics data on the inserter's addressee files.

6. The system of claim 4, further comprising means at the data center for processing the addressee files of inserter entities to add demographics data and for transmitting the processed addressee files back to the inserters.

7. The system of claim 4, further comprising means at the data center for arranging payment to inserter entities for including in envelopes documents originating at particular publishing entities.

8. In a mailing processing method, the steps comprising:

(a) interconnecting via a network, plural independent mailers of advertising documents having intelligent inserters for inserting documents into envelopes for mail distribution to a plurality of addressees, plural independent publishers of advertising documents, and a data center having data processing facilities,

(b) compiling a database of addressees and a database of addressee profiles and a database of documents to be distributed,

(c) determining which documents from the plural publishers are relevant to which addressees, and

(d) transmitting to the mailers and to the publishers information enabling the mailers to receive from the publishers those documents which can be combined into a single mailing to a particular addressee.

9. In the method of claim 8, including the steps of compiling a database of excess envelope capacity available at particular mailers for particular addressees, transmitting via the network to the publishers data representative of the excess envelope capacity available at mailers whereby the publishers can utilize said excess envelope capacity for their own documents.

* * * * *

US006049775A

United States Patent [19]

Gertner et al.

[11] Patent Number: 6,049,775

[45] Date of Patent: Apr. 11, 2000

[54] **SYSTEMS, METHODS AND COMPUTER PROGRAM PRODUCTS FOR MONITORING AND CONTROLLING MAIL PROCESSING DEVICES**

[75] Inventors: **Patricia A. Gertner**, Raleigh; **William A. Sterling**, Cary; **Richard C. Nardin**, Apex, all of N.C.

[73] Assignee: **Bell & Howell Mail and Messaging Technologies Company**, Durham, N.C.

[21] Appl. No.: **09/079,620**

[22] Filed: **May 15, 1998**

Related U.S. Application Data

[63] Continuation-in-part of application No. 09/016,715, Jan. 30, 1998.

[60] Provisional application No. 60/085,479, May 14, 1998.

[51] **Int. Cl.**[7] **G06F 17/60**; G05B 15/02

[52] **U.S. Cl.** **705/8**; 345/326; 345/333; 345/334; 700/9; 700/11; 700/17; 700/95; 700/100

[58] **Field of Search** 345/326, 333, 345/334, 348; 364/138, 140.01, 146, 188, 191, 192, 400, 468.01, 468.06, 468.09; 705/1, 7, 8, 9; 700/9, 11, 17, 95, 100

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,522,588	8/1970	Clarke, Jr. et al.	340/825.06
3,582,621	6/1971	Lawler	364/138
4,346,446	8/1982	Erbstein et al.	702/182
4,517,637	5/1985	Cassell	364/138
4,752,950	6/1988	Le Carpentier	379/106.11
4,798,040	1/1989	Haas et al.	53/460.77
4,837,701	6/1989	Sansone et al.	705/405
5,305,055	4/1994	Ebner et al.	399/9
5,309,556	5/1994	Sismilch	345/349
5,470,218	11/1995	Hillman et al.	425/144
5,471,399	11/1995	Tanaka et al.	364/491
5,628,249	5/1997	Cordery et al.	101/91
5,684,706	11/1997	Harman et al.	705/406 X
5,812,394	9/1998	Lewis et al.	364/146
5,859,778	1/1999	Kuroda et al.	364/474.11
5,870,698	2/1999	Riedel et al.	702/182

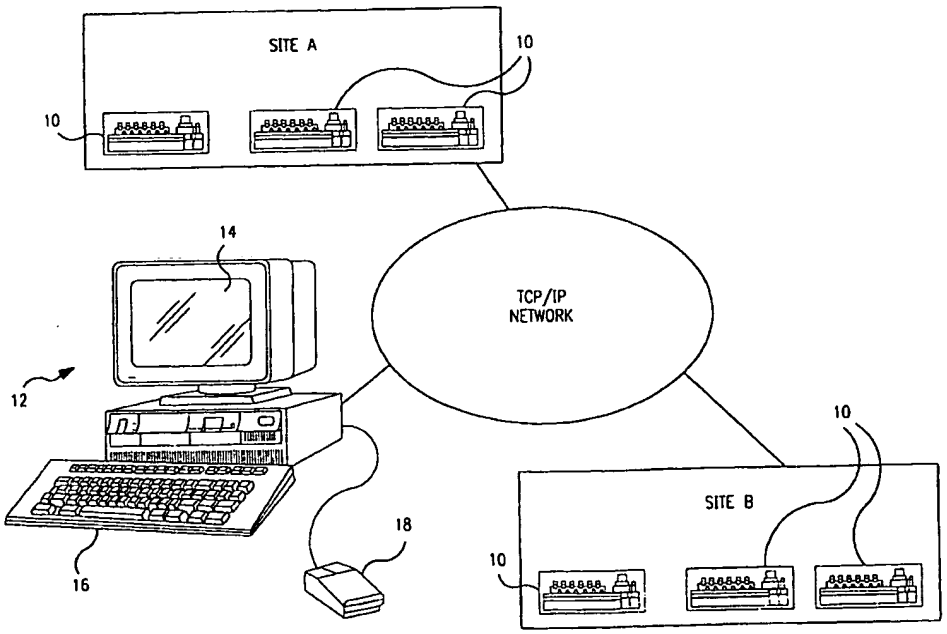
Primary Examiner—Edward R. Cosimano

Attorney, Agent, or Firm—Jenkins & Wilson, P.A.

[57] **ABSTRACT**

A system, method, device, and/or computer program product for remotely managing a plurality of remote mail processing devices. The system comprises receiving means for receiving operation related information from each of said plurality of remote mail processing devices. The system also includes monitoring means for remotely monitoring operation related information from each of said plurality of remote mail processing devices. Features of the present invention include the ability to create threshold settings for separate mail processing devices and individual jobs running on the separate mail processing devices. Violations of the threshold settings will result in an alarm indicating deviation from pre-set job goals. A user also has the ability to remotely create, maintain, and communicate valid mail processing device operator lists for specified mail processing devices. A run-tag editor is included which allows the user remotely access a stored database of completed job information in order to correct any job identification errors that may have been input by a mail processing device operator.

54 Claims, 17 Drawing Sheets



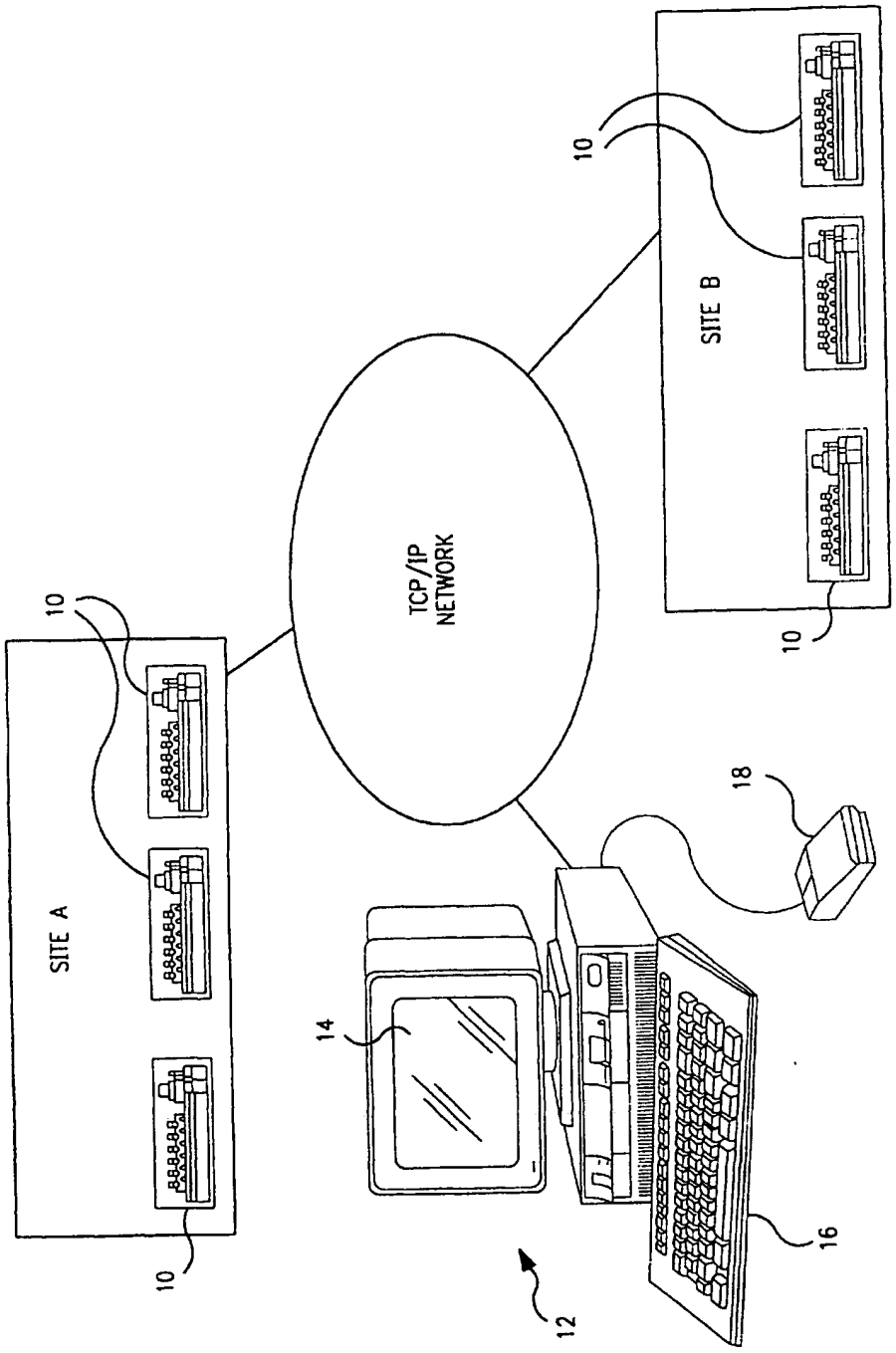


FIG. 1

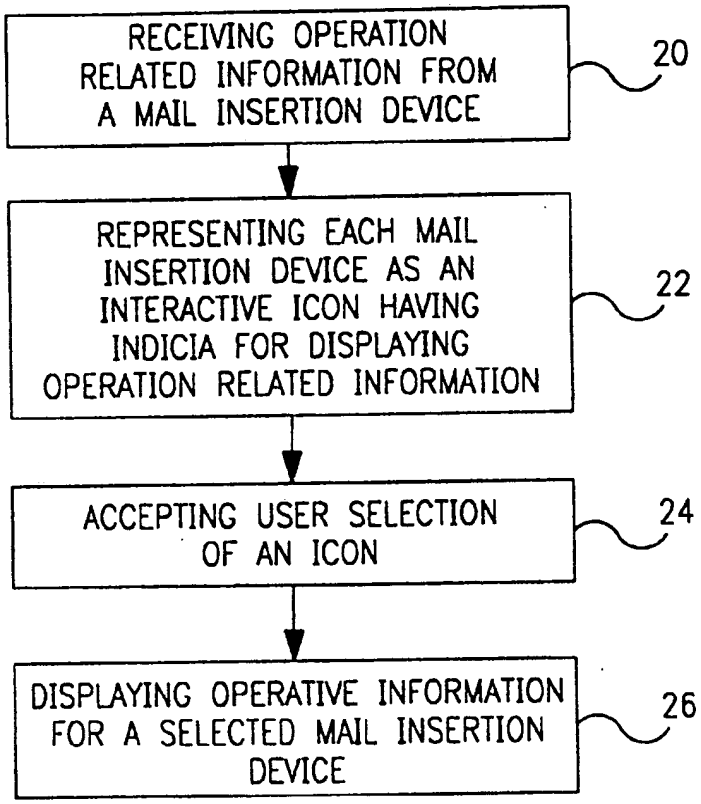

RECEIVING OPERATION RELATED INFORMATION FROM A MAIL INSERTION DEVICE
20
REPRESENTING EACH MAIL INSERTION DEVICE AS AN INTERACTIVE ICON HAVING INDICIA FOR DISPLAYING OPERATION RELATED INFORMATION
22
ACCEPTING USER SELECTION OF AN ICON
24
DISPLAYING OPERATIVE INFORMATION FOR A SELECTED MAIL INSERTION DEVICE
26

FIG. 2

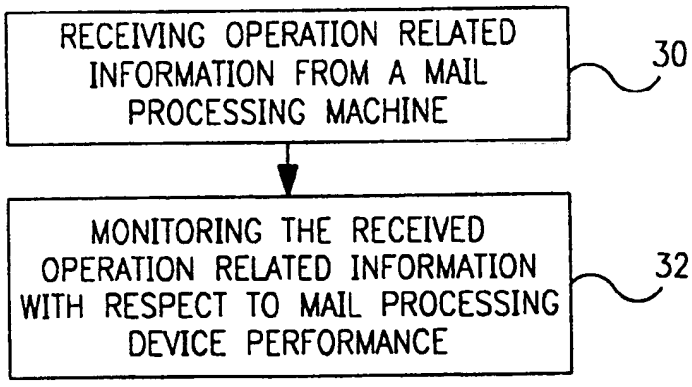

RECEIVING OPERATION RELATED INFORMATION FROM A MAIL PROCESSING MACHINE
30
MONITORING THE RECEIVED OPERATION RELATED INFORMATION WITH RESPECT TO MAIL PROCESSING DEVICE PERFORMANCE
32

FIG. 3

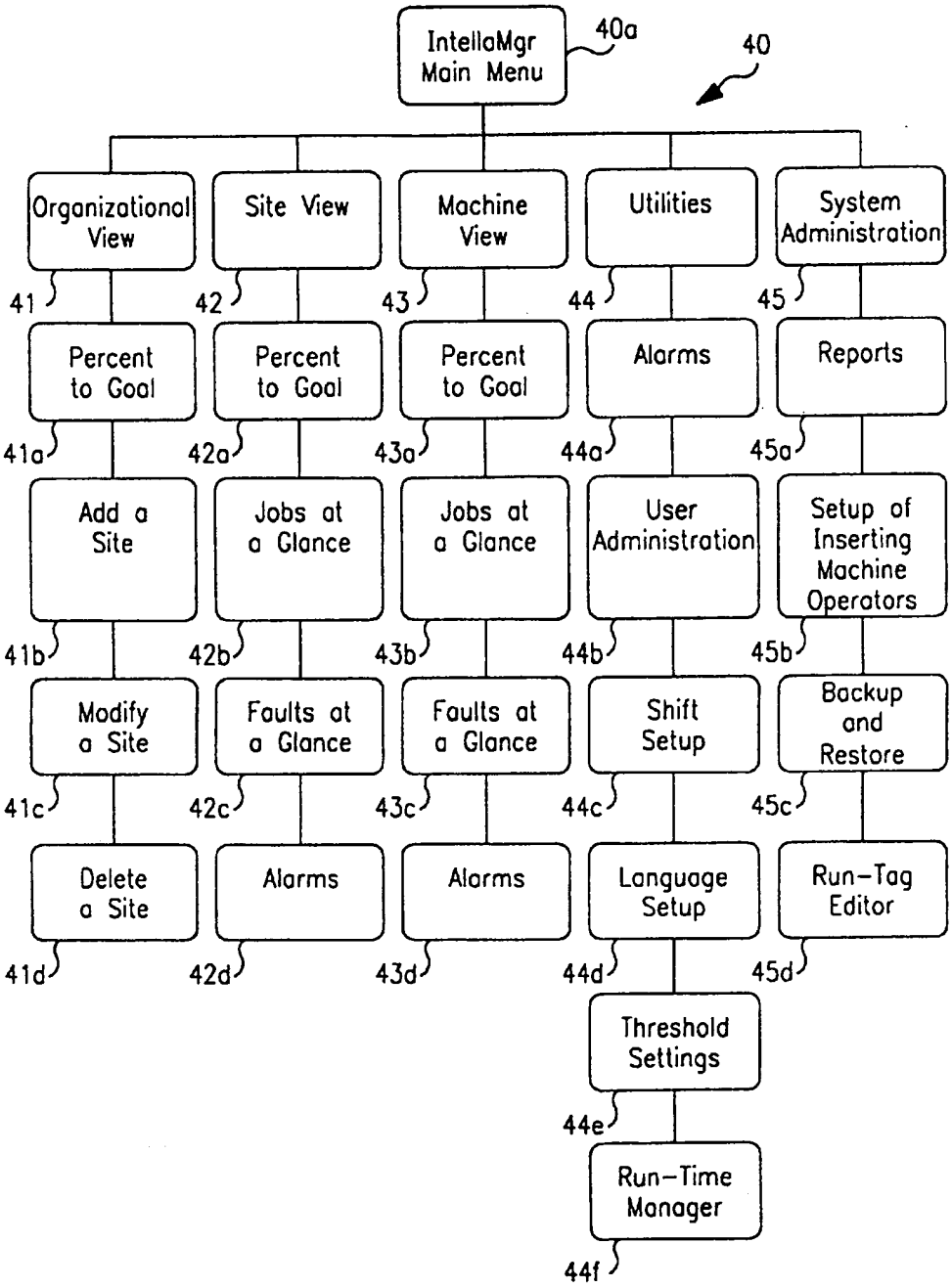

IntellaMgr Main Menu
40a
40
Organizational View
Site View
Machine View
Utilities
System Administration
41
42
43
44
45
Percent to Goal
Percent to Goal
Percent to Goal
Alarms
Reports
41a
42a
43a
44a
45a
Add a Site
Jobs at a Glance
Jobs at a Glance
User Administration
Setup of Inserting Machine Operators
41b
42b
43b
44b
45b
Modify a Site
Faults at a Glance
Faults at a Glance
Shift Setup
Backup and Restore
41c
42c
43c
44c
45c
Delete a Site
Alarms
Alarms
Language Setup
Run-Tag Editor
41d
42d
43d
44d
45d
Threshold Settings
44e
Run-Time Manager
44f

FIG. 4

FIG. 5

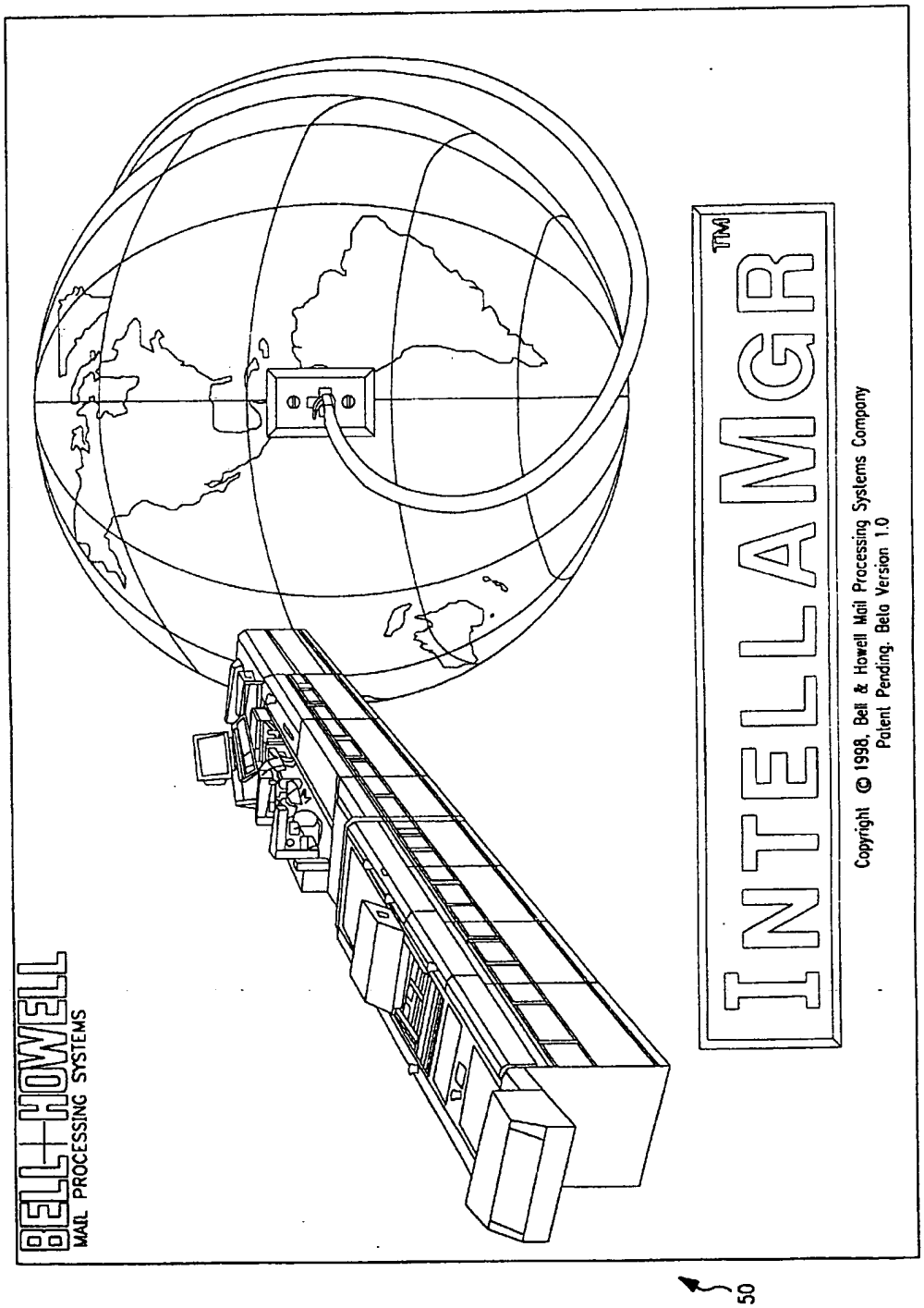

BELL+HOWELL
MAIL PROCESSING SYSTEMS
INTELLAMGR™
Copyright © 1998, Bell & Howell Mail Processing Systems Company
Patent Pending. Beta Version 1.0
50

FIG. 6

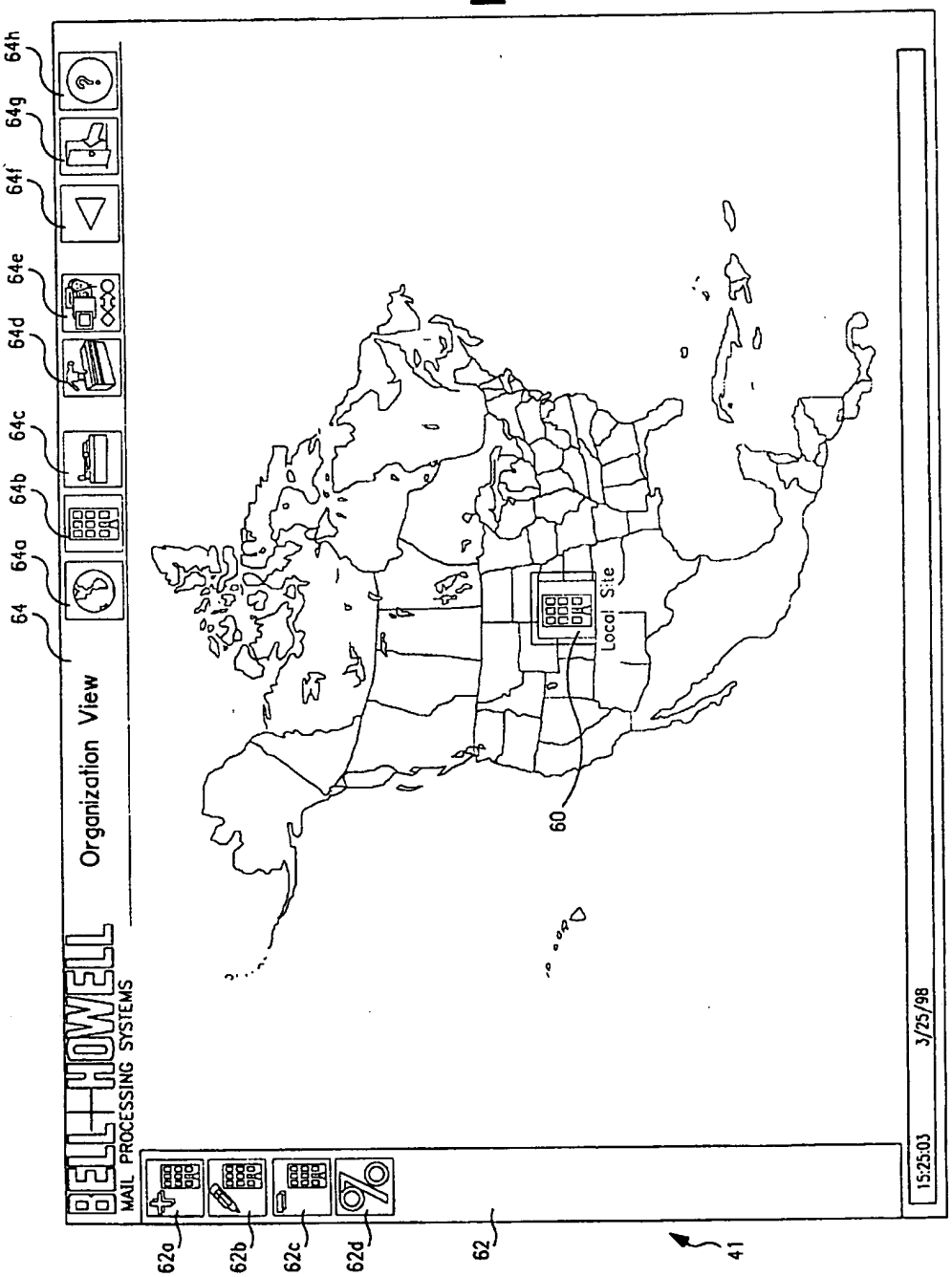

BELL+HOWELL
MAIL PROCESSING SYSTEMS
Organization View
Local Site
60
64
64a
64b
64c
64d
64e
64f
64g
64h
62a
62b
62c
62d
62
41
15:25:03
3/25/98

FIG. 7

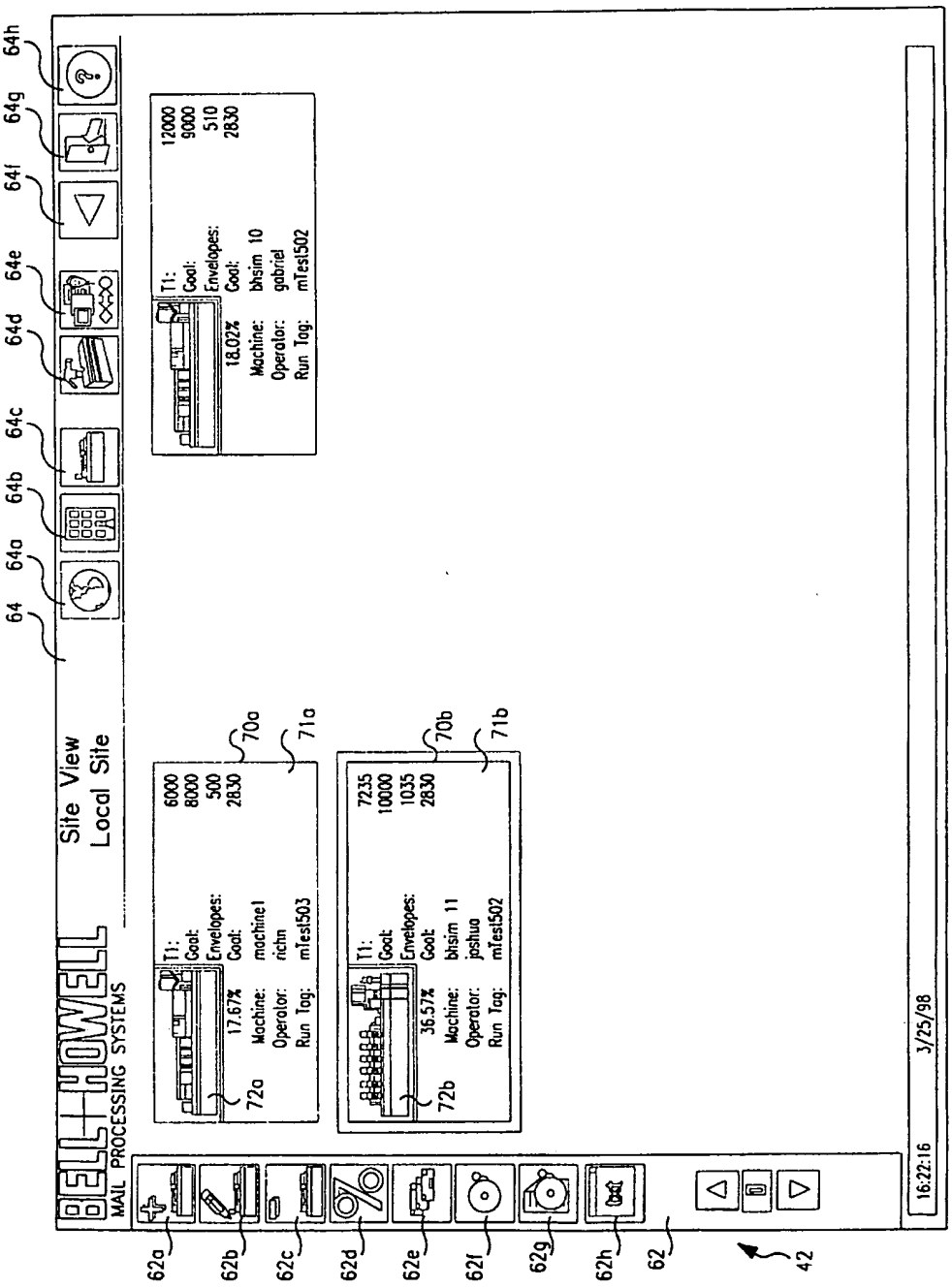
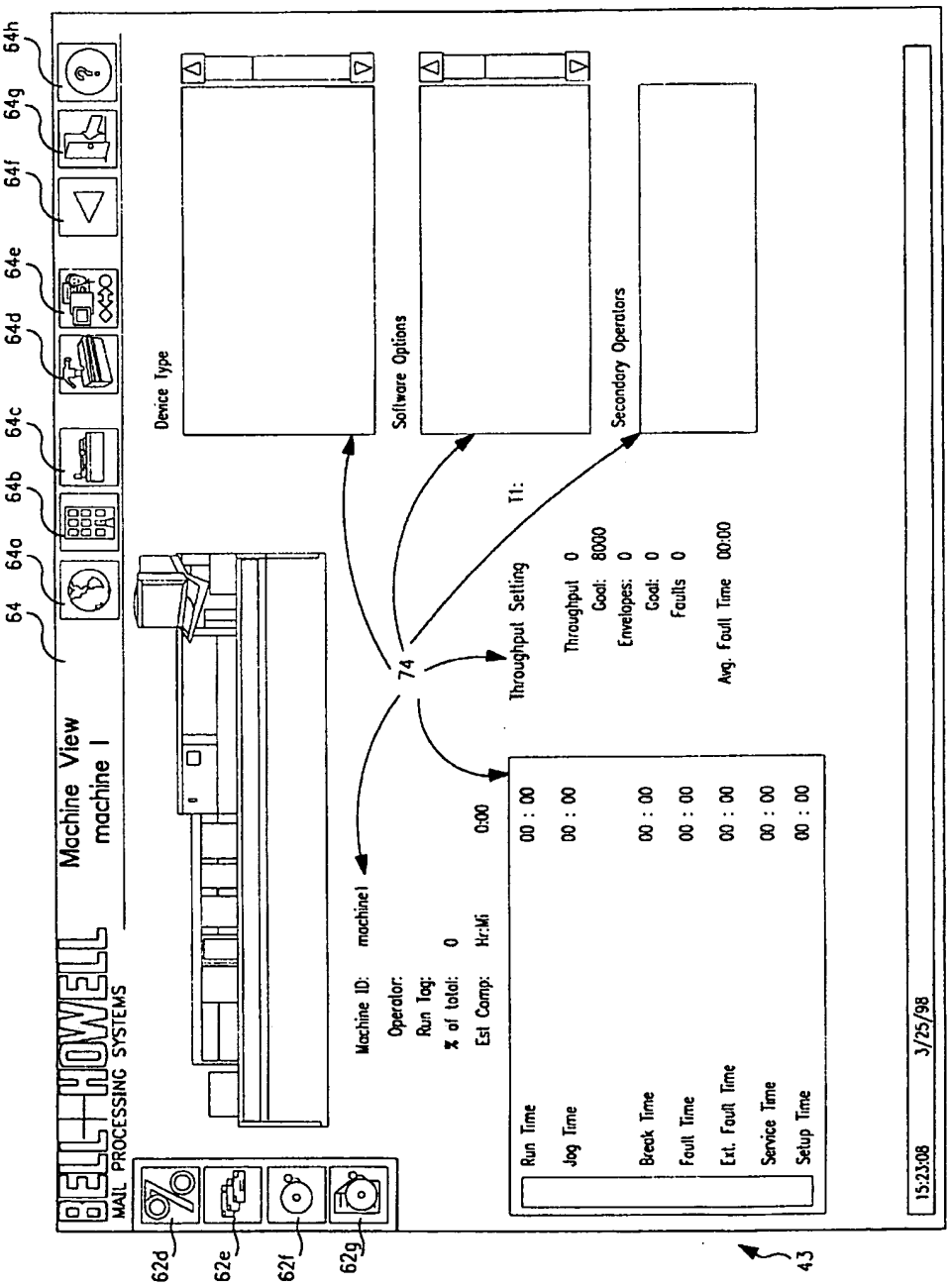

BELL+HOWELL
MAIL PROCESSING SYSTEMS
Site View
Local Site
64
64a
64b
64c
64d
64e
64f
64g
64h
T1:
Goal: 6000
8000
Envelopes: 500
Goal: 2830
17.67%
Machine: machine1
Operator: richn
Run Tag: mTest503
70a
71a
72a
T1:
Goal: 7235
10000
Envelopes: 1035
Goal: 2830
36.57%
Machine: bhsim 11
Operator: joshua
Run Tag: mTest502
70b
71b
72b
T1:
Goal: 12000
9000
Envelopes: 510
Goal: 2830
18.02%
Machine: bhsim 10
Operator: gabriel
Run Tag: mTest502
16:22:16
3/25/98
62a
62b
62c
62d
62e
62f
62g
62h
62
42

FIG. 8


BELL+HOWELL
MAIL PROCESSING SYSTEMS
Machine View
machine 1
64
64a
64b
64c
64d
64e
64f
64g
64h
Device Type
Software Options
Secondary Operators
74
T1:
Throughput Setting
Throughput 0
Goal: 8000
Envelopes: 0
Goal: 0
Faults 0
Avg. Fault Time 00:00
Machine ID: machine1
Operator:
Run Tag:
% of total: 0
Est Comp: Hr:Mi 0:00
Run Time 00 : 00
Jog Time 00 : 00
Break Time 00 : 00
Fault Time 00 : 00
Ext. Fault Time 00 : 00
Service Time 00 : 00
Setup Time 00 : 00
15:23:08 3/25/98
62d
62e
62f
62g
43

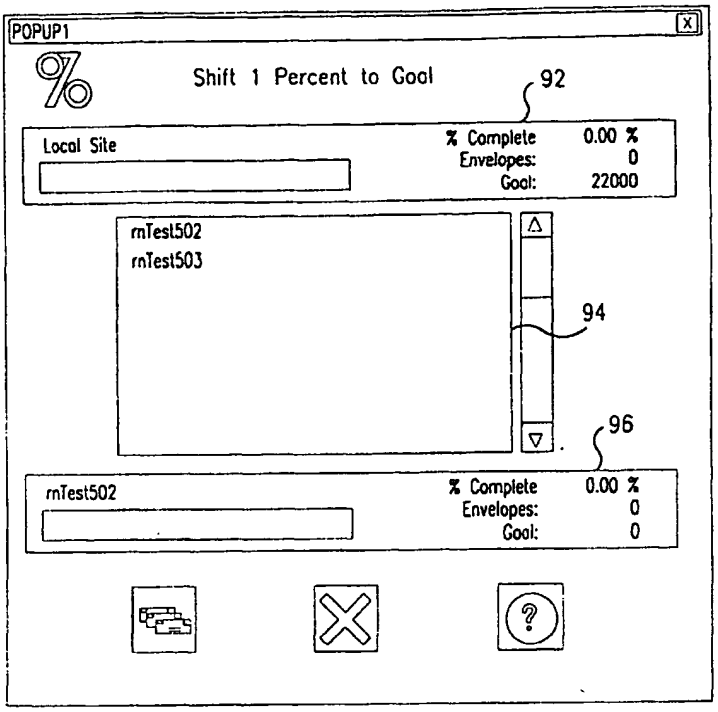

POPUP1
Shift 1 Percent to Goal
92
Local Site
% Complete 0.00 %
Envelopes: 0
Goal: 22000
rnTest502
rnTest503
94
96
rnTest502
% Complete 0.00 %
Envelopes: 0
Goal: 0
90

FIG. 9

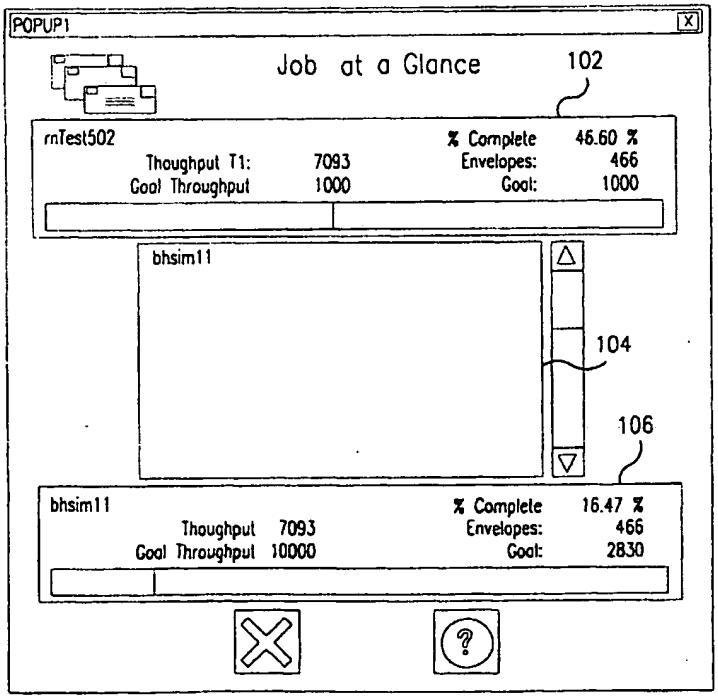

POPUP1
Job at a Glance
102
rnTest502
% Complete 46.60 %
Thoughput T1: 7093
Envelopes: 466
Goal Throughput 1000
Goal: 1000
bhsim11
104
106
bhsim11
% Complete 16.47 %
Thoughput 7093
Envelopes: 466
Goal Throughput 10000
Goal: 2830
100

FIG. 10

FIG. 11

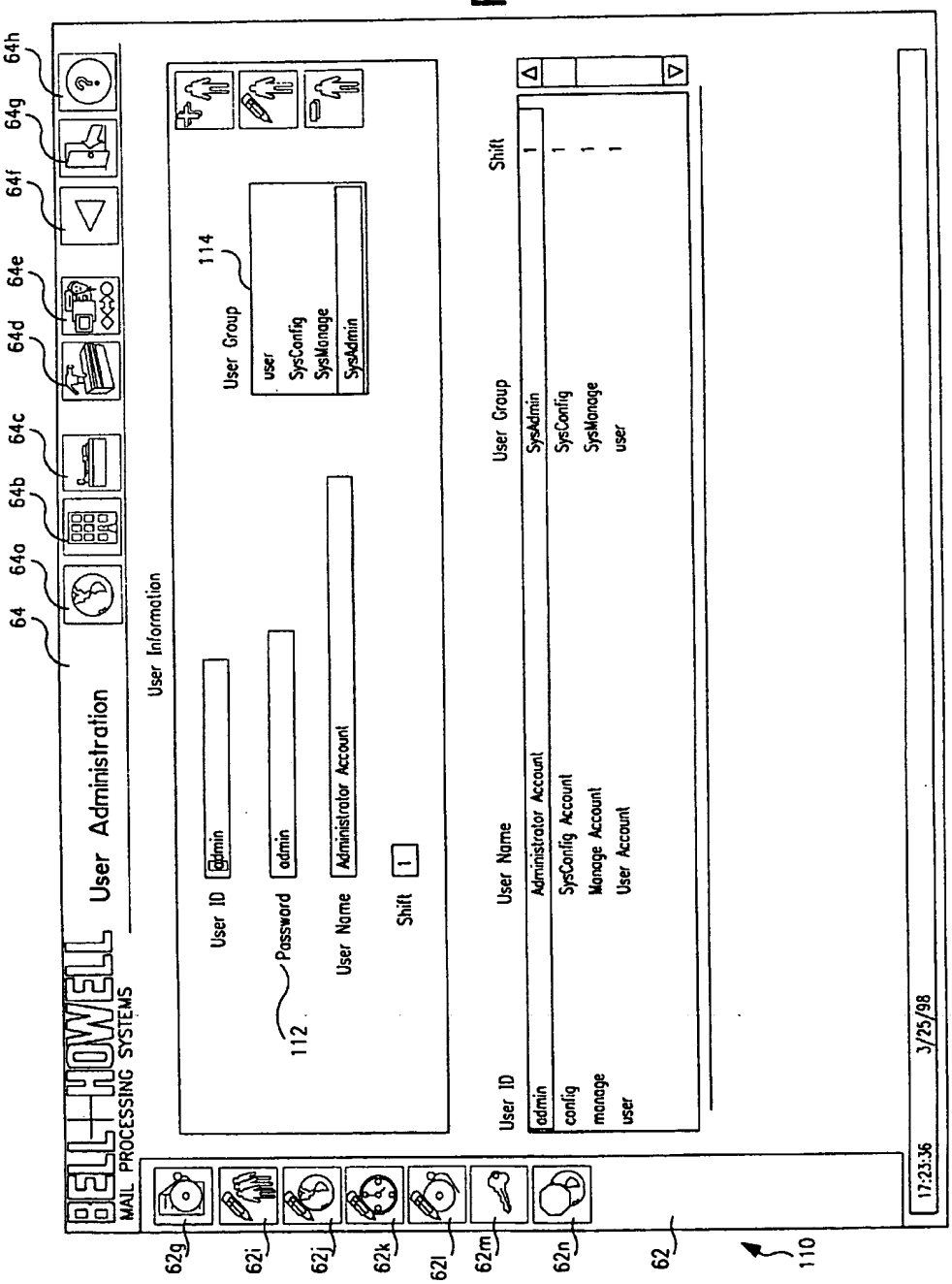

BELL+HOWELL
MAIL PROCESSING SYSTEMS
User Administration
User Information
User ID
admin
Password
admin
User Name
Administrator Account
Shift
1
User Group
user
SysConfig
SysManage
SysAdmin
114
112
User ID
User Name
User Group
Shift
admin
Administrator Account
SysAdmin
1
config
SysConfig Account
SysConfig
1
manage
Manage Account
SysManage
1
user
User Account
user
1
17:23:36
3/25/98
64
64a
64b
64c
64d
64e
64f
64g
64h
62g
62i
62j
62k
62l
62m
62n
62
110

FIG. 12

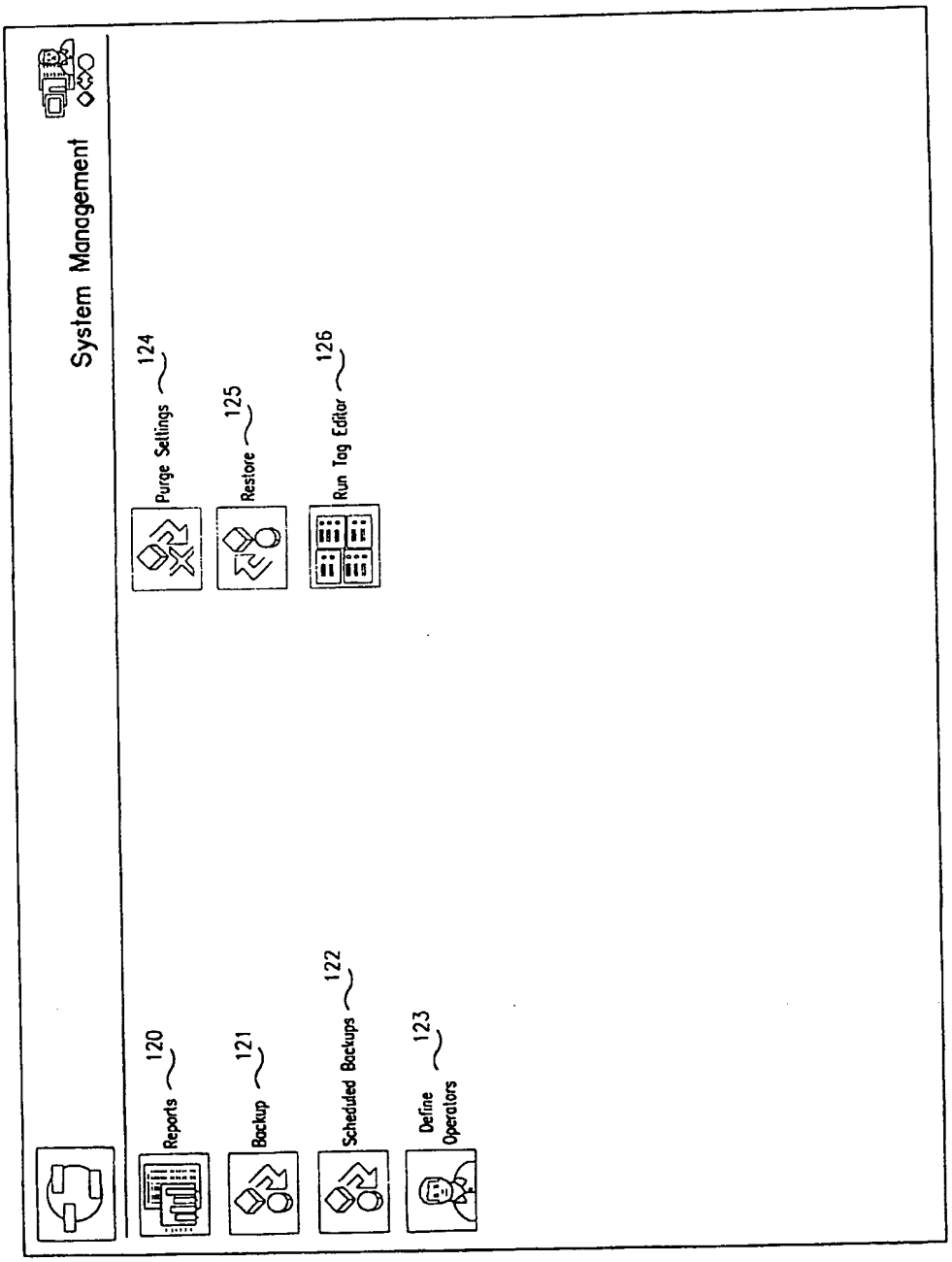

System Management
Purge Settings
124
Restore
125
Run Tag Editor
126
Reports
120
Backup
121
Scheduled Backups
122
Define Operators
123

FIG. 13

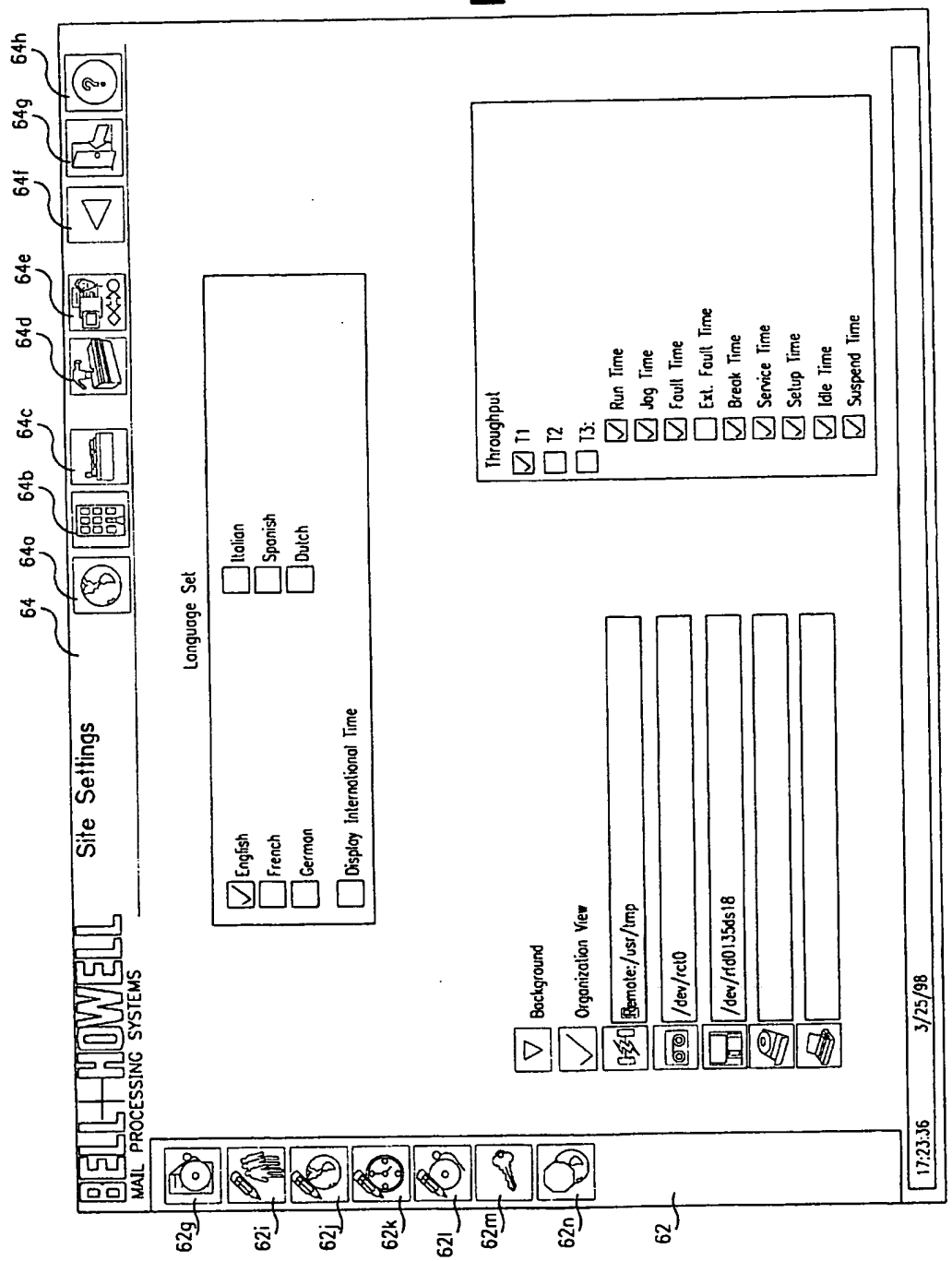

BELL+HOWELL
MAIL PROCESSING SYSTEMS
Site Settings
Language Set
English
French
German
Italian
Spanish
Dutch
Display International Time
Throughput
T1
T2
T3:
Run Time
Jog Time
Fault Time
Ext. Fault Time
Break Time
Service Time
Setup Time
Idle Time
Suspend Time
Background
Organization View
Remote:/usr/tmp
/dev/rct0
/dev/rd01.35ds18
17:23:36
3/25/98
64
64a
64b
64c
64d
64e
64f
64g
64h
62
62g
62i
62j
62k
62l
62m
62n

FIG. 14

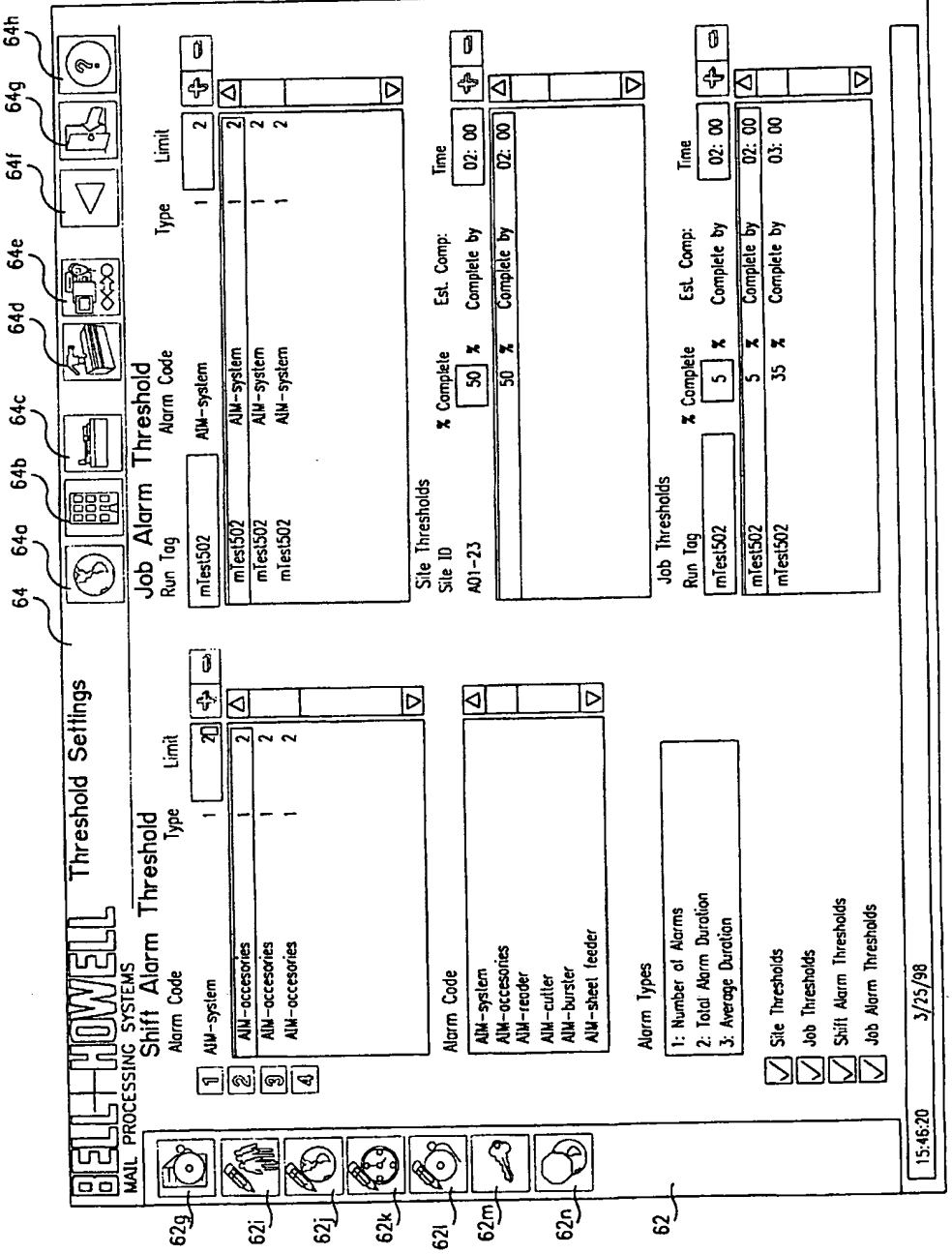

BELL-HOWELL
MAIL PROCESSING SYSTEMS
Threshold Settings
Shift Alarm Threshold
Alarm Code
Type
Limit
AIM-system
AIM-accesories
AIM-reader
AIM-cutter
AIM-burster
AIM-sheet feeder
Job Alarm Threshold
Run Tag
mTest502
% Complete
Est. Comp:
Complete by
Time
02: 00
03: 00
Site Thresholds
Site ID
A01-23
Job Thresholds
Alarm Types
1: Number of Alarms
2: Total Alarm Duration
3: Average Duration
Site Thresholds
Job Thresholds
Shift Alarm Thresholds
Job Alarm Thresholds
15:46:20
3/25/98
64
64a
64b
64c
64d
64e
64f
64g
64h
62g
62i
62j
62k
62l
62m
62n
62

FIG. 15

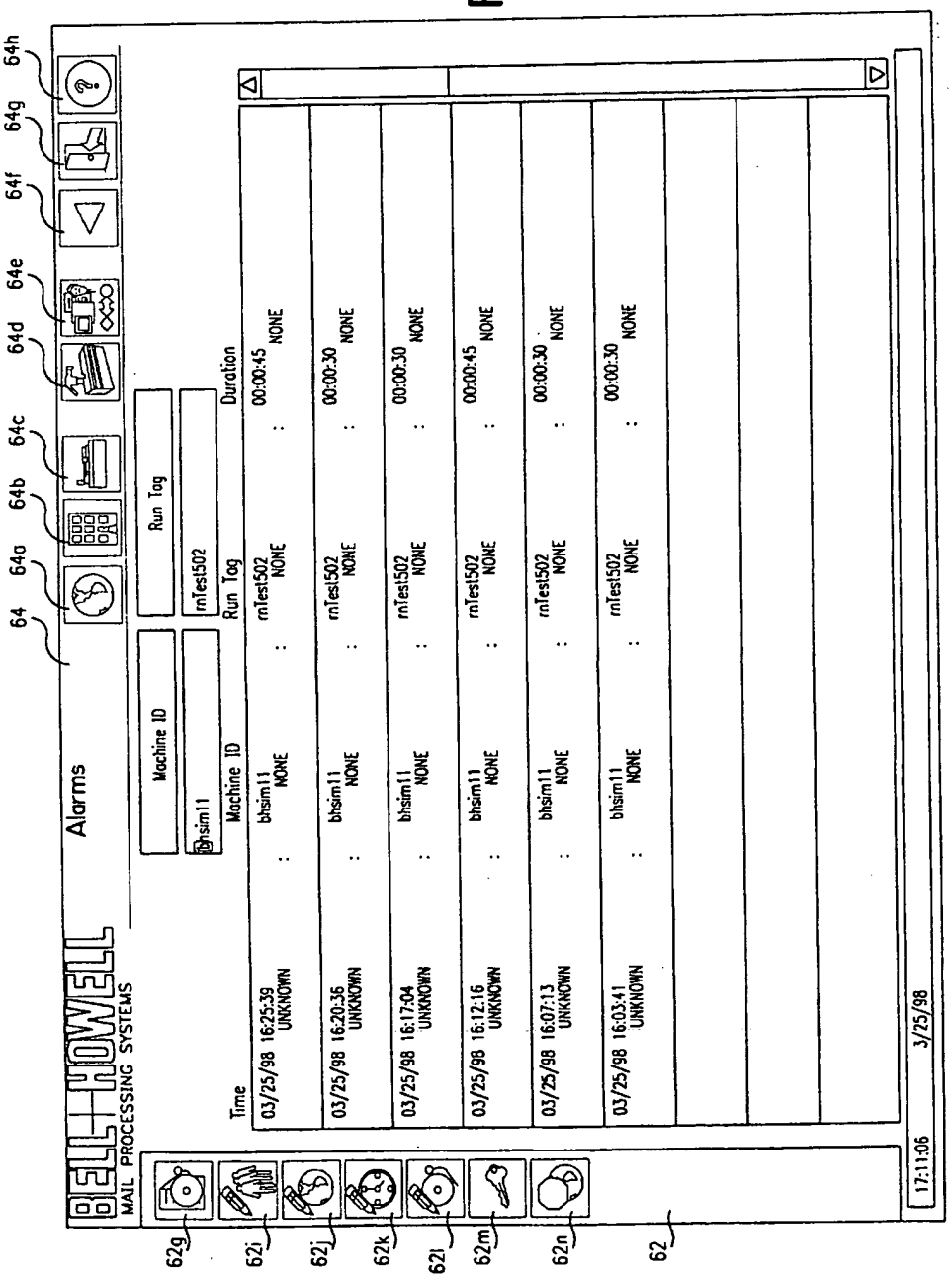

BELL+HOWELL
MAIL PROCESSING SYSTEMS
Alarms
Machine ID
bhsim11
Run Tag
rnTest502
Time
Machine ID
Run Tag
Duration
03/25/98 16:25:39 bhsim11 rnTest502 00:00:45
UNKNOWN NONE NONE NONE
03/25/98 16:20:36 bhsim11 rnTest502 00:00:30
UNKNOWN NONE NONE NONE
03/25/98 16:17:04 bhsim11 rnTest502 00:00:30
UNKNOWN NONE NONE NONE
03/25/98 16:12:16 bhsim11 rnTest502 00:00:45
UNKNOWN NONE NONE NONE
03/25/98 16:07:13 bhsim11 rnTest502 00:00:30
UNKNOWN NONE NONE NONE
03/25/98 16:03:41 bhsim11 rnTest502 00:00:30
UNKNOWN NONE NONE NONE
17:11:06
3/25/98
64
64a
64b
64c
64d
64e
64f
64g
64h
62g
62i
62j
62k
62l
62m
62n
62

FIG. 16

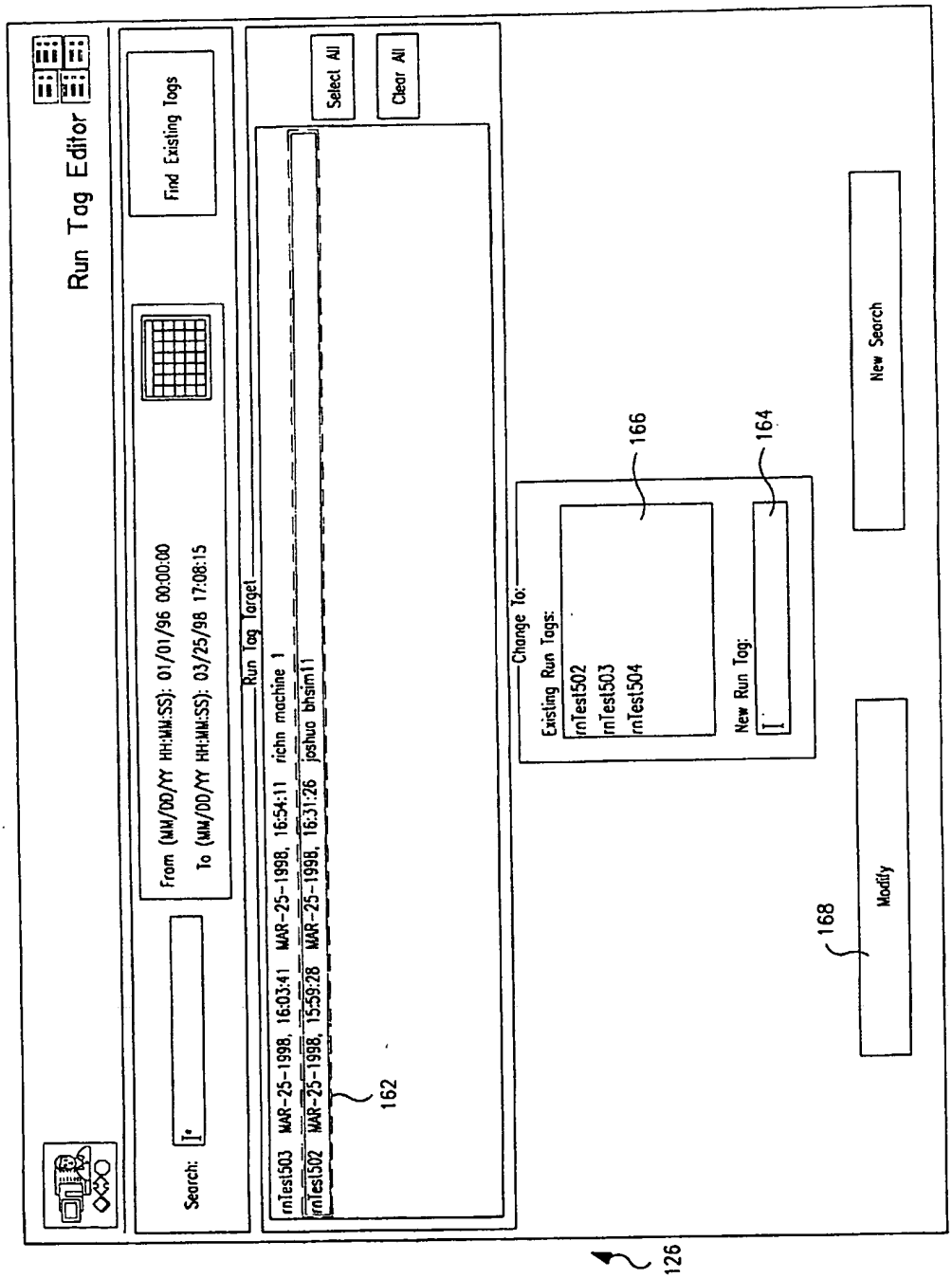

Run Tag Editor
Search:
From (MM/DD/YY HH:MM:SS): 01/01/96 00:00:00
To (MM/DD/YY HH:MM:SS): 03/25/98 17:08:15
Find Existing Tags
Run Tag Target
rmTest503 MAR-25-1998, 16:03:41 MAR-25-1998, 16:54:11 richn machine 1
rmTest502 MAR-25-1998, 15:59:28 MAR-25-1998, 16:31:26 joshuo bhsim11
Select All
Clear All
162
Change To:
Existing Run Tags:
rmTest502
rmTest503
rmTest504
166
New Run Tag:
164
New Search
Modify
168
126

FIG. 17

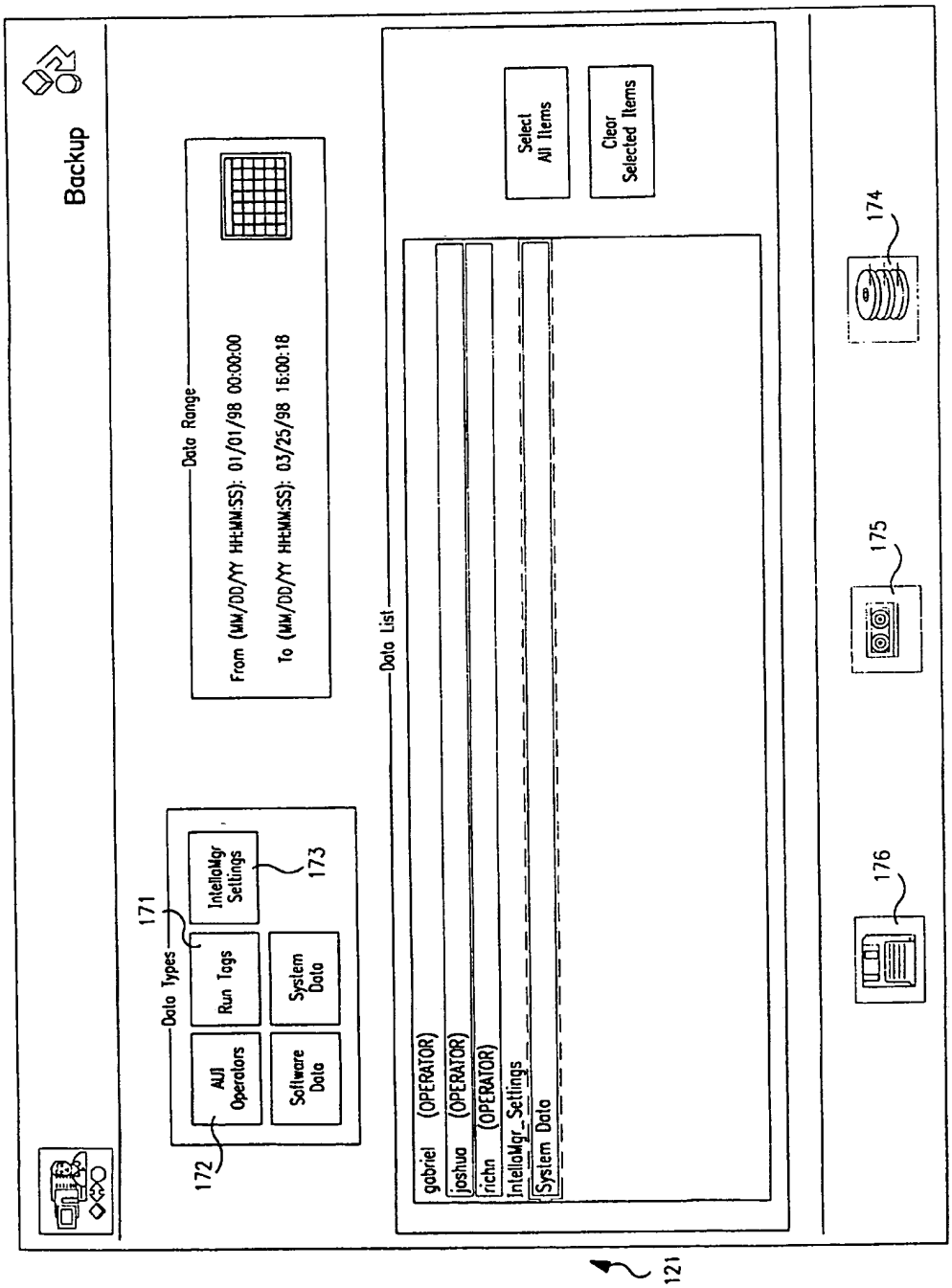

Backup
Data Range
From (MM/DD/YY HH:MM:SS): 01/01/98 00:00:00
To (MM/DD/YY HH:MM:SS): 03/25/98 16:00:18
Select All Items
Clear Selected Items
Data Types
All Operators
Run Tags
IntelliMgr Settings
Software Data
System Data
Data List
gabriel (OPERATOR)
joshua (OPERATOR)
richn (OPERATOR)
IntelliMgr_Settings
System Data
171
172
173
174
175
176
121

FIG. 18

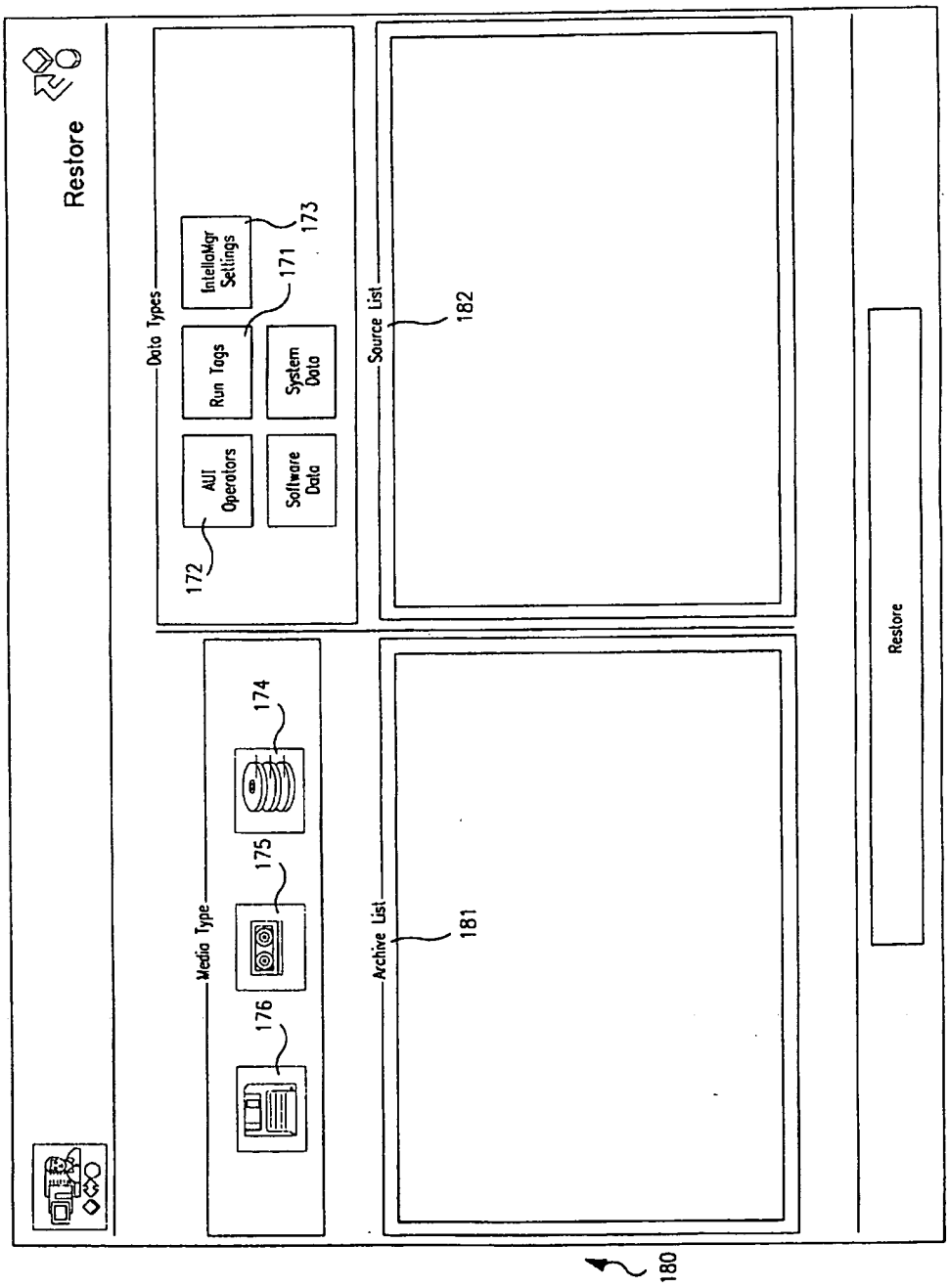
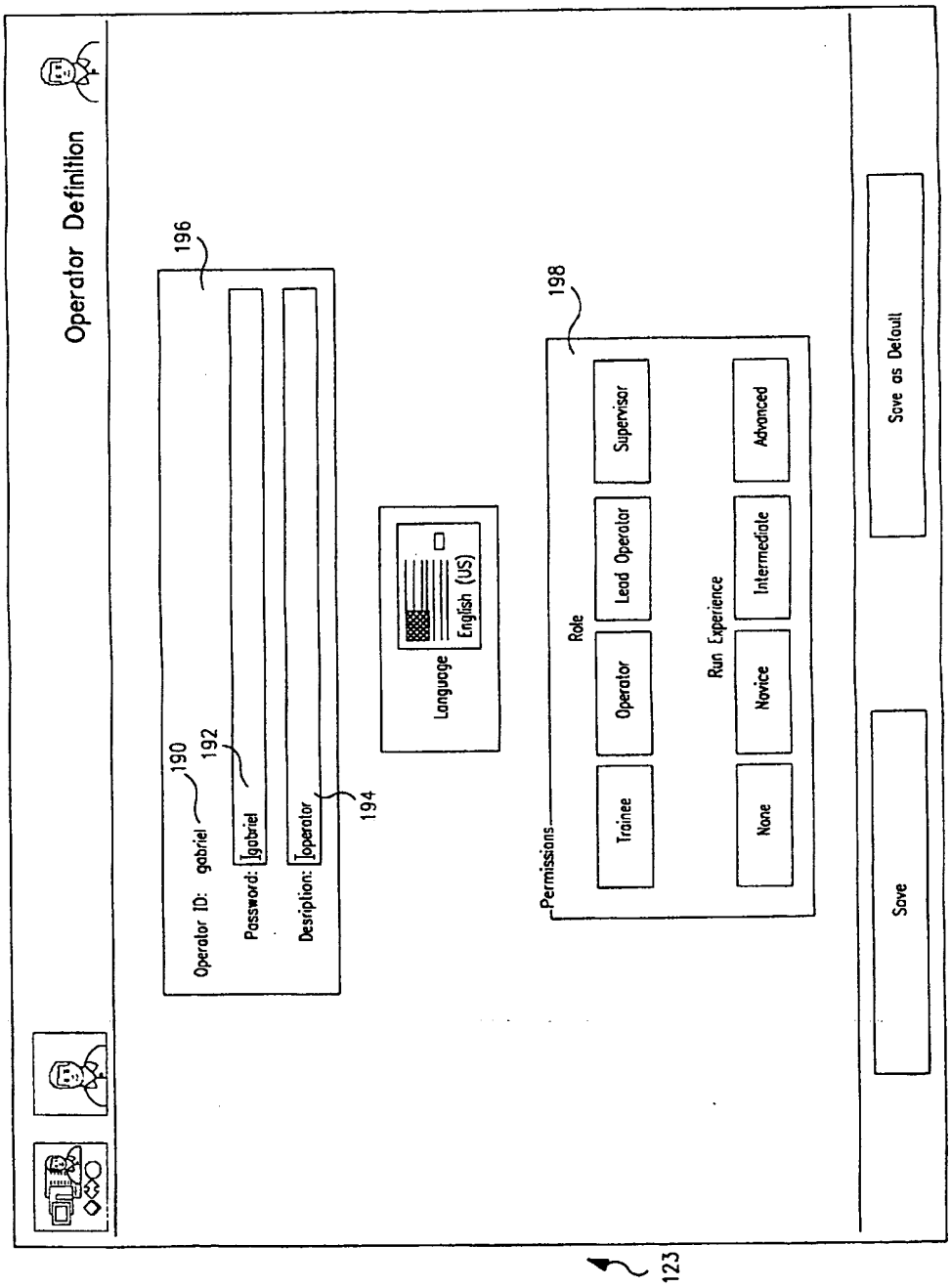

Restore
Data Types
IntelliMgr Settings
Run Tags
System Data
AUI Operators
Software Data
Source List
Media Type
Archive List
Restore
171
172
173
174
175
176
180
181
182

FIG. 19


Operator Definition
196
Operator ID: gabriel
190
192
Password: gabriel
Description: Operator
194
Language
English (US)
198
Permissions
Role
Trainee
Operator
Lead Operator
Supervisor
Run Experience
None
Novice
Intermediate
Advanced
Save
Save as Default
123

SYSTEMS, METHODS AND COMPUTER PROGRAM PRODUCTS FOR MONITORING AND CONTROLLING MAIL PROCESSING DEVICES

RELATED APPLICATIONS

The present application is a Continuation-In-Part Application of, and claims the benefit under 37 C.F.R. §1.53(b) of, application Ser. No. 09/016,715 entitled "Systems, Methods and Computer Program Products for Monitoring and Controlling Mail Processing Devices", filed on Jan. 30, 1998 (pending). The present application also claims priority and benefit under 35 U.S.C. §119(e) of U.S. Provisional Patent Application Ser. No. 60/085,479 entitled "Systems and Method for Monitoring and Controlling Mail Processing Devices", filed on May 14, 1998. Both of the aforementioned patent applications are hereby incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to mail processing equipment and more particularly to automated mail processing systems. Specifically, the present invention relates to remote monitoring and control of mail processing systems and the features thereof.

BACKGROUND OF THE INVENTION

Customized, high volume mail processing systems are used by financial institutions, insurance companies, public utilities, and various other businesses to prepare and mail letters and packages. Mail processing systems include mail inserting systems, sorting systems, and a range of modular attachments for increasing the productivity of large scale mail production operations. Mail insertion systems include a device known as a mail inserter, which physically "stuffs" individual envelopes with the appropriate contents. Many businesses, such as financial institutions and public utilities, often produce voluminous mailings on a routine basis. As such, mailings are often performed at various locations to reduce the load on any given location and to be closer to the destination of the mailings.

Unfortunately, monitoring and controlling remotely-located high volume mailing systems with existing control systems may be difficult. Existing control systems may not readily facilitate monitoring multiple remotely-located mailing systems from a central location. Furthermore, existing control systems may not readily facilitate monitoring various operating performance indicators of individual mail inserter devices within multiple remotely-located mailing systems.

DISCLOSURE OF THE INVENTION

It is therefore an object of the present invention to facilitate the monitoring of remotely-located high volume mailing systems from a central location.

It is another object of the present invention to facilitate the monitoring of various operating performance indicators of individual mail inserter devices within multiple remotely-located mailing systems.

Further objects of the invention will become apparent to those skilled in the art with reference to the accompanying figures and written description below.

In accordance with the present invention, systems, methods, and computer program products for managing and controlling, from a central location, a plurality of remotely situated mail processing devices are provided. Operation related information is displayed pertaining to each of said plurality of remotely situated mail processing devices such as, for instance, a mail insertion machine. Each mail insertion machine is represented on a computer visual display as an interactive icon. Visible indicia are associated with each icon for conveying, in real time, operational status information pertaining to the mail inserting machine represented by the icon.

The indicia associated with the mail insertion machine icons change appearance in response to a change in the certain operation related information received from a respective mail insertion machine. The indicia can be displayed as a color selected from a plurality of colors, wherein each color represents a respective operational condition or status of the mail insertion machine. Each interactive mail insertion machine site is displayable as an icon in a manner depicting the geographical location of each mail insertion device as it relates to a larger area, such as, for instance, the continental United States, or South America. Interactive icons may be, inter alia, added, deleted, and their location and appearance on the display can be modified.

One embodiment of the present invention describes a system for remotely managing a plurality of remote mail processing devices. The system comprises receiving means for receiving operation related information from each of the remote mail processing devices. The system also includes monitoring means for remotely monitoring operation related information from each of the remote mail processing devices operatively connected to the receiving means.

According to another aspect of the present invention, each interactive icon is responsive to user actions for displaying selective operation related information about a respective mail insertion device. Each interactive icon accepts user selections via an input device. When the user activates an interactive icon, various operational information about the mail insertion device represented by the icon is displayed. This operational information may include machine status, fault information, and various other types of information related to the operation and performance of the mail insertion device. Information about the operator of the mail insertion device may also be available.

The present invention is advantageous because users can be provided with the ability to monitor multiple mail insertion systems from a single location not necessarily proximate to the machines being monitored. Furthermore, the various stages of the insertion process can be monitored in real time. The present invention can facilitate managing automated mail processing systems to increase productivity and decrease costs associated with insertion. The present invention is also advantageous because a list of mail processing device operators can be created, modified, and sent out from a central location.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing advantages and features of the present invention will be appreciated more fully from the following description with reference to the accompanying drawings in which:

FIG. **1** of the drawings is a schematic diagram of a system for monitoring multiple mail processing systems, in accordance with the present invention;

FIG. **2** of the drawings schematically illustrates, in block diagram form, operations for managing a plurality of mail insertion devices, according to the present invention;

FIG. **3** of the drawings schematically illustrates, in block diagram form, operations for controlling a plurality of mail insertion devices, according to the present invention;

FIG. 4 of the drawings schematically illustrates a hierarchical arrangement of computer screens for managing a plurality of mail insertion devices, according to the present invention;

FIG. 5 of the drawings illustrates a startup screen presented to the user upon startup of the present invention;

FIG. 6 of the drawings illustrates an Organizational View screen according to the present invention;

FIG. 7 of the drawings illustrates a Site View screen according to the present invention;

FIG. 8 of the drawings illustrates a Machine View screen according to the present invention;

FIG. 9 of the drawings illustrates a Percent to Goal screen according to the present invention;

FIG. 10 of the drawings illustrates a Job at a Glance screen according to the present invention;

FIG. 11 of the drawings illustrates a User Administration screen according to the present invention;

FIG. 12 of the drawings illustrates a System Management screen according to the present invention;

FIG. 13 of the drawings illustrates a pop-up screen for selecting a language for text to be displayed in;

FIG. 14 of the drawings illustrates a Threshold Setting screen according to the present invention;

FIG. 15 of the drawings illustrates a pop-up screen for Alarm Status;

FIG. 16 of the drawings illustrates a Run Tag Editor screen according to the present invention;

FIG. 17 of the drawings illustrates a Backup screen according to the present invention;

FIG. 18 of the drawings illustrates a Restore screen according to the present invention; and

FIG. 19 of the drawings illustrates an Operator Definition screen according to the present invention.

BEST MODE FOR CARRYING OUT THE INVENTION

The present invention now is described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

As will be appreciated by one of skill in the art, the present invention may be embodied as a method, data processing system, or computer program product. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of a computer program product on a computer-readable storage medium having computer-readable program code means embodied in the medium. Any suitable computer readable medium may be utilized including hard disks, CD-ROMs, optical storage devices, or magnetic storage devices.

The present invention is described below with reference to flowchart illustrations of methods, apparatus (systems) and computer program products according to the invention. It will be understood that each block of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These computer program instructions may be loaded onto a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the computer or other programmable data processing apparatus create means for implementing the functions specified in the flowchart block or blocks. These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

Accordingly, blocks of the flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions and program instruction means for performing the specified functions. It will also be understood that each block of the flowchart illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by special purpose hardware-based computer systems which perform the specified functions or steps, or combinations of special purpose hardware and computer instructions.

As will readily be appreciated by those skilled in the art, mail processing systems can encompass several types of remote and/or local mail processing machines, including, but not limited to, mail insertion machines and mail sorting machines. In a conventional mail processing system, mail inserting machines (also referred to hereinafter as insertion machines or insertion devices) are utilized for stuffing inserts into awaiting envelopes. Thereafter, the stuffed envelopes are sealed and various other operations associated with mail processing are performed. Insertion machines typically operate at high speeds and can often stuff ten thousand (10,000) envelopes or more per hour. An exemplary insertion machine is described in Applicants' issued U.S. Pat. No. 4,798,040 to Haas et al., the disclosure of which is incorporated herein by reference.

Referring now to FIG. 1, a schematic diagram of multiple mail insertion machines monitored and controlled by computer systems, methods and computer program products, in accordance with the present invention, is illustrated. Sites A and B each have one or more mail insertion machines **10** and are monitored and controlled via the present invention operating on computer system **12**. Computer system **12** includes a display **14** for visually displaying information to a computer user, a central processing unit (CPU) and internal memory. Keyboard **16** and mouse **18** allow the user to provide input to the central processing unit. With the mouse **18**, the user may move a pointer on the display **14** to an area displaying an object. By pressing and releasing a button on the mouse **18** while the pointer is in the area displaying the object or icon, the user "activates" the icon. This is also referred to hereinafter as "clicking on" an icon. Computer system **12**, serves as data processing means for transmitting and receiving information from and to each mail insertion machine **10** and displaying same via display **14**. Other

computing systems, including mainframe computing systems, may be utilized to facilitate communicating with the various mail processing devices being monitored and controlled.

The present invention may be stored in computer system **12**, either within internal memory or on internal disk storage. The present invention may also be stored on computer readable media and used therefrom by computer system **12**. As will be described in detail below, the present invention alters the operation of computer system **12**, allowing it to monitor and control a plurality of mail insertion devices in real time.

Referring now to the block diagram of FIG. **2**, operations for managing a plurality of mail insertion devices, according to one aspect of the present invention, are illustrated. Operations include receiving operation related information from mail insertion devices (Block **20**); representing each of the mail insertion devices as an interactive icon having indicia for displaying operation related information for a respective mail insertion device (Block **22**); accepting selection of an interactive icon (Block **24**); and displaying operating status for the selected mail insertion device (Block **26**).

Operation related information includes, but is not limited to, organizational, site, machine, utility, and system administration information. Organizational information comprises site addition, deletion, and modification capability as well as instantaneous production and goal data. Site information comprises instantaneous production and goal data, job data, fault data, and alarm data with respect to a particular site. Machine information comprises instantaneous production and goal data, job data, fault data, and alarm data with respect to a particular machine. Utility information comprises alarm data, user administration data, shift data, language data, threshold setting data, and run-time manager data. System administration information comprises report data, remote machine setup data, backup and restore data, and run tag editor data.

It is to be understood that a user does not necessarily have to select an icon to view information about a particular mail insertion device. Various information relating to a mail insertion device may be automatically updated and displayed. The operations illustrated in FIG. **2** may be performed in the various "views" described in detail below using FIGS. **4–18**.

Referring now to FIG. **3**, operations for controlling a plurality of mail insertion devices, according to one aspect of the present invention, are illustrated. Operations include receiving operation related information from mail processing devices (Block **30**); and monitoring the received operation related information with respect to mail processing device performance (Block **32**).

Referring now to FIG. **4**, a hierarchical arrangement **40** of computer screens for facilitating management of a plurality of mail insertion devices, according to the present invention, is illustrated. From a main menu **40***a* presented to the user on a computer display, the user may select an "Organizational View" **41** of multiple sites having mail insertion machines, a "Site View" **42** of individual sites having mail insertion machines, a "Machine View" **43** of individual insertion machines at a particular site; a "Utilities" **44** selection comprising several options; and a "System Administration" **45** selection comprising several options.

Various screens containing operating information are available from each of the respective Organizational View **41**, Site View **42**, and Machine View **43** screens. For example, the user may access the "Percent to Goal" screen **41***a*, and the "Add" **41***b*, "Modify" **41***c*, or "Delete" **41***d* settings screens from the Organizational View screen **41**. The user may access the "Percent to Goal" screen **42***a*, the "Jobs at a Glance" screen **42***b*, the "Faults at a Glance" screen **42***c*, and the "Alarms" screen **42***d* from the Site View Screen **42**. The user may access the "Percent to Goal" screen **43***a*, the "Jobs at a Glance" screen **43***b*, the "Faults at a Glance" screen **43***c*, and the "Alarms" screen **43***d* from the Machine View Screen **43**. Each of these screens, and the information presented thereby, are described in detail below.

The user may select (i.e., run) various utility programs via screens accessible via the "Utilities" screen **44** such as the "Alarms" screen **44***a*, the "User Administration" screen **44***b*, the "Shift Setup" screen **44***c*, the "Language Setup" screen **44***d*, the "Threshold Settings" screen **44***e*, and the "Run-Time Manager" screen **44***f*. In addition, the user may perform various administrative tasks via the "Systems Administration" screen **45** such as the "Reports" screen **45***a*, the "Setup of Inserting Machine Operators" screen **45***b*, the "Backup/Restore" screen **45***c*, and the "Run-Tag Editor" screen **45***d*. Each of these screens, and the information presented thereby, are described in detail below.

Logo Screen and Log-In

The initial screen on the user's display, when the present invention is initiated on a computer, is preferably a logo or start-up screen **50** such as that illustrated in FIG. **5**. Clicking at any point on the start-up screen **50** preferably activates a login procedure via a pop-up screen. For security reasons, it is required that the user enter a user name and password in order to proceed. As would be understood by those having skill in the art, options available to a user upon login may depend on permissions assigned to the particular user.

Once properly logged in, the present invention allows users to view various operational and performance information about multiple remotely-located mail insertion machines. Users are presented with a hierarchy of views depicting organizational, site, and machine-specific perspectives of their mail insertion machines. The user can display text throughout the various screens in a variety of languages including, but not limited to, English, Spanish, French, Italian, German and Dutch. Preferably, a language can be selected via a pop-up screen such as that illustrated in FIG. **13**. Each hierarchical view will be described below in detail.

Each view screen, organizational, site and machine, also presents the user with various toolbars containing icons for performing various functions. In the embodiment of the Organizational View Screen illustrated in FIG. **6**, two toolbars **62** and **64** are presented to the user. Each toolbar contains icons for performing specific functions. Preferably, toolbars **62** and **64** are presented to the user with the same configuration in all views in order to maintain a consistent and easy to use interface. It is understood, however, that certain functions may not be executable from certain screens. In this case, the icons representing those functions are shaded or greyed out indicating they are not a valid selection or operation from the present screen.

Table 1 below describes the function of each of the icons contained within toolbar **62**.

TABLE 1

Icon #	Icon Name	Icon Function
62a	Add (Site/Machine)	Add a site or a machine.
62b	Edit (Site/Machine)	Edit a selected site or machine.
62c	Delete (Site/Machine)	Delete a selected site or machine.

TABLE 1-continued

Icon #	Icon Name	Icon Function
62d	Percent to Goal	Access the Percent to Goal panel.
62e	Job at a Glance	Access the Job at a Glance panel.
62f	Faults at a Glance	Access the Faults at a Glance panel.
62g	Alarms	Access the Alarms panel.
62h	Text	Toggle between a graphic and text display.
62i	User Administration	Access the User Administration panel.
62j	Site Settings	Access the Site Settings panel.
62k	Shift Settings	Access the Shift Settings panel.
62l	Threshold Settings	Access the Threshold Settings panel.
62m	Run Manager	Access the Run Manager panel.
62n	Shut-down Application	Shut-down the current application.

Table 2 below describes the function of each of the icons contained within toolbar **64**.

TABLE 2

Icon #	Icon Name	Icon Function
64a	Organizational View	Display all sites currently configured.
64b	Site View	Display the site settings for a selected site.
64c	Machine View	Display all the machines defined at a particular site.
64d	Utilities	Display the Alarms panel. Access other Utilities.
64e	System Administration	Access various system administration tasks.
64f	Back	Return to the previous panel.
64g	Log off	Log out and return to main panel.
64h	Help	Access help panels.

It is understood that additional graphic designs may be utilized for each icon without departing from the spirit and intent of the present invention.

Organizational View

The Organizational View screen **41**, illustrated in FIG. **6**, preferably includes a geographical background on which the user can place up to twelve (12) sites having mail insertion machines in a geographically oriented arrangement. Although the illustrated embodiment depicts North America as the background, it is to be understood that any geographical depiction is possible using the present invention. Furthermore, no geographical background may be presented if so desired by the user. In the case where users do not have multiple sites, the organizational view can be deactivated and removed from the screen hierarchy.

The Organizational View screen **41** allows the user to display site icons **60** for each site having mail insertion machines. The user will be able to activate site icons **60** and be advanced to "Site View" for the respective site, as described in detail below.

The user can add, delete and modify information associated with each site by "clicking on" or otherwise activating the respective site icon **60** and then clicking on or otherwise activating the respective add, edit, or delete icons **62***a*, **62***b*, **62***c*. If the user is modifying a site, a pop-up window with the selected site's information appears on the screen for editing when icon **62***b* is activated. If the user is deleting a site, the site icon will simply disappear from the screen when icon **62***c* is activated. To add a site, the user activates icon **62***a* and a pop-up window, designed to receive input from the user, appears on the screen.

Site View

Preferably, when the user "clicks on" or otherwise activates a site icon, a Site View screen is presented to the user for the site represented by the activated icon. A Site View screen displays the location of mail insertion machines at a selected site via machine icons. An exemplary Site View screen **42** is illustrated in FIG. **7**, wherein two insertion machines are represented by machine icons **70***a* and **70***b*. Preferably a maximum of twelve (12) mail insertion machines are displayed per Site View screen **42**, not to exceed a total of one hundred (100) machines.

For each mail insertion machine at a site, the following information is preferably available via a respective machine icon: machine status, actual and goal throughput, actual and goal envelope counts, percent completion for the current run, the current operator identification, and the current job identification. Preferably, all mail insertion machine information is available and displayable in real time. It is understood that the term "real time" shall mean a minimal refresh rate of any screen of 30 seconds or less.

The operational status or condition of a particular mail insertion machine is discernable at all times by the color of indicia surrounding each machine icon. Accordingly, the user does not have to activate the machine icon to determine the operational status of a respective mail insertion machine. As illustrated in FIG. **7**, each machine icon **70***a*, **70***b* has an area of indicia **71***a*, **71***b* immediately surrounding the graphic portion **72***a*, **72***b* of each machine icon that changes colors depending on the operating conditions of the particular insertion machine. Exemplary operating conditions and their respective colors are presented in Table 3 below.

TABLE 3

Insertion Machine Operating Condition	Color Surrounding Machine Icon
Off	Background Color
Stopped	Grey
Running/Run Pending	Green
Jogging/Jog Pending	Yellow
Faulted/Insertion Machine Communications Failure	Red
Hold/Track Hold	Orange
Operator Break	Pink
Service	White

The present invention is not limited to color as a visual indication of insertion machine operational status. Operational status can be indicated by other visual schemes which change the display of an icon in response to actual machine operational condition changes. Machine status information available from Site View includes, but is not limited to, actual and goal throughput, actual and goal envelope counts, the percent complete for a current run, the current operator identification, and the current job identification number. The information that is displayed is the current information for the inserting machine and shows the progress being made on the job that is currently running on that machine.

Users in Site View can also arrange a series of machine icons to represent the physical layout of mail insertion machines. Icons can represent the type of insertion machine and indicate the type of communication protocols being utilized. The user can add, modify, and delete insertion machine icons at each site. When adding an insertion machine, the user is prompted for the following information: type of insertion machine, machine identification, machine name, and various network parameters.

Machine View

When the user double clicks or otherwise activates an interactive machine icon in the Site View screen of FIG. **7** the Machine View screen **43** of FIG. **8** is presented for the mail insertion machine represented by that machine icon. Using the Machine View screen **43**, the user is able to view

various operational information **74** including, but not limited to, the following: machine status, actual and goal throughput, actual and goal envelope counts, estimated time to completion, the currently running job identification, the current fault count, the average fault length, a bar graph of time usage, device information, software package information and operator information. In addition, the user will be able to view the current T1, T2 or T3 formula results for each machine. As is known to those with skill in the art, for Advanced User Interface (AUI) type insertion machines (manufactured by Bell & Howell Mail Processing Systems, Allentown, Pa., and Durham, N.C.), T1 is defined as a throughput calculation of "envelopes per hour" and is calculated using the following equation:

(run envelopes/(run time+fault time))*3600

T2 is defined as a calculation of "percent utilization" and is calculated using the following equation:

(total envelopes*100)/(theoretical throughput)

For both T1 and T2, the following definitions apply:

total envelopes=run envelopes+jog envelopes

theoretical throughput=(average run speed*(run time+jog time+fault time))/3600

average run speed=run machine cycles/(run time/3600)

For Inserting System Machine Connection (ISMC) insertion machines (manufactured by Bell & Howell Mail Processing Systems) T1 is defined as follows:

T1=((total envelopes−service envelopes)*3600)/(run time+fault time)

As is known by those skilled in the art, T2 is not calculated for ISMC insertion machines because the ISMC interface generally does not supply machine cycle information.

For both types of insertion machines, T3 is a user-defined formula, and can be configured in Site Settings for each site. The numerator is"run envelopes" and the denominator may be a sum of user selected time measurements. The user may select any or all of the following time measurements to be included in the denominator: run time, jog time, fault time, extended fault time, break time, service time, idle time, or suspend time. An exemplary T3 calculation is:

(run envelopes/(run time+jog time+break time))*3600

Remote Site Connectivity

The user can view other site information by going to the Organizational View of FIG. **6** and selecting another site's icon **64***b*. The other site icon must have been previously configured in order to be present on the screen. Once the user has selected a remote site icon **64***b*, the Site View, Machine View, "Percent to Goal" pop-up, "Job at a Glance" pop-up and "Faults at a Glance" pop-up will now display information from the remote site. The user is prohibited from changing setup information for the remote site. For instance, the user can not change thresholds or run tag goals for the remote site.

Other Screens and Windows

The following screens and pop-up windows are presented to the user when the user activates certain icons and/or automatically when certain events occur.

"Faults at a Glance" Pop-Up

Icons for providing fault and job information are preferably provided in both Machine View and Site View. The user can select this pop-up window (not shown) by clicking on the "Faults at a Glance" button **62***f* on the left side of the Site View screen of FIG. **7** or the Machine View screen of FIG. **8**.

"Percent to Goal" Pop-Up

A Percent to Goal pop-up window **90**, illustrated in FIG. **9**, appears when the user clicks on the "%" icon **62***d* in either Machine View of FIG. **8** or Site View of FIG. **7**. The Percent to Goal pop-up permits the user to view current statistics. The top portion **92** displays all information pertaining to an entire site. Statistics for jobs currently running or completed during the current shift on the insertion machines at the site may be included in the top portion **92**. A list **94** of jobs allows the user to select a job and view a summary of the selected job in the bottom portion **96** of the window **90**.

"Job at a Glance" Pop-Up

A Job at a Glance pop-up window **100**, illustrated in FIG. **10**, appears when the user clicks on the "Job at a Glance" button **62***e* on the side of various screens. The top section **102** shows a summary of all elements of a job. The list **104** in the center portion of the window **100** contains all insertion machines running that particular job. When the user selects one of those insertion machines, details about the selected insertion machine will appear in the bottom section **106**. Preferred information displayed in the bottom section **106** includes, but is not limited to: machine identification, operators running machine(s), current throughput, goal throughput, current accounts completed, and total accounts in job.

User Administration

The User Administration utility screen **110**, illustrated in FIG. **11**, allows the user to add, edit, and delete users, including changing the security group of which the user is a member. In the illustrated embodiment, the top left portion **112** of the screen **110** is used to configure a user's account. In the illustrated embodiment, the top right portion **114** of the screen **110** is used to configure security groups of users. Each security group preferably has a name and a set of designated privileges. The types of privileges include: status view, operator view, reports, alarm view, file transfer, edit configuration, job programming, maintain system, and edit users.

System Management and Utilities

FIG. **12** illustrates the"System Management" screen in which the user is provided with various system management tools including report generation **120**, data backup **121**, scheduled data backup **122**, operator setup **123**, purge settings **124**, data restore **125**, and run tag editor **126**. Various utilities may also be provided for performing a variety of functions including configuring user accounts, changing the display language, and configuring shift setups. The user can view individual occurrences of alarms from the mail insertion machines as well as set alarm thresholds for specific jobs. Alarm messages preferably can be filtered either by job identification, insertion machine identification or both.

Thresholds are a significant feature of the present invention. Thresholds allow the user to pro-actively monitor the progress of inserting machines and take corrective actions if necessary. There are two different types of thresholds. The first type is alarm thresholds. Alarm thresholds (FIG. **14**) are based on occurrences of errors within a particular inserting machine. The second type is production thresholds. Production thresholds are based on the progress of completed envelopes compared to the expected goal of completed envelopes for a particular job.

Alarm Thresholds

The user may set either Shift or Run tag thresholds (FIG. **15**) of three (3) different types: number of alarm occurrences, total duration of the alarms, or average duration of the alarms. The user may select to apply the threshold to a group of alarms for a shift or for a specific run tag. When an inserting machine crosses the threshold, the user is alerted with a pop-up which displays the violated threshold.

Example:

For Shift 1, a threshold is set for five occurrences of "AIM—reader" errors. If an inserting machine exhibits five of these errors, the user is alerted via an alert pop-up. Once the alert pop-up is dismissed, the count is re-initialized to zero and the threshold monitoring is resumed.

In addition, a pop-up window is presented to the user for viewing occurrences of thresholds on groups of errors. Table 4 below illustrates some of the errors that may be monitored and logged.

TABLE 4

General Error Type	Specific Error Type
Aim	System
Aim	Accessories
Aim	Reader
Aim	Cutter
Aim	Burster
Aim	Sheet Feeder
Aim	Turnover/Sequencer
Aim	Hold Module
Aim	Right Angle Turn
Aim	Accumulator
Aim	Diverter
Aim	Folder
Aim	Collector
Aim	Interface
Aim	Slitter
Aim	Other
Aim	All
Insert Feeder	System
Insert Feeder	Reader
Insert Feeder	Miss
Insert Feeder	Double
Insert Feeder	Other
Insert Feeder	All
Envelope Feeder	System
Envelope Feeder	Reader
Envelope Feeder	Miss
Envelope Feeder	Double
Envelope Feeder	Other
Envelope Feeder	All
Base Machine	—
Finishing	Turnover
Finishing	Sealer
Finishing	Bridge
Finishing	Other
Finishing	All
Delivery	Modular Delivery
Delivery	Postage Meter
Delivery	Printer
Delivery	Output Device
Delivery	Other
Delivery	All

Preferably, a pop-up window automatically appears on the user's display when a threshold is crossed, in order to alert the user to the particular condition. Information that is preferably displayed in such a window includes: identification of insertion machine having an error, error type, number of occurrences of error, total errors, total time lost, and average reset time. Preferably, the item that caused the threshold to trigger the pop-up window will be highlighted. For example, if a threshold was set for an average reset time of two (2) minutes and the average reset time calculated equals three (3) minutes, the "Threshold Violation" pop-up window would appear on the user's display with the average reset time of three (3) minutes highlighted. If the user needs to see specific error occurrences, he/she can go to an Alarm Browser and filter the errors by machine identification or job identification.

It is understood that additional alarm thresholds may be utilized without departing from the spirit and intent of the present invention.

Production Thresholds

The user may set either Site or Run tag thresholds for production levels. The user selects a "percentage to be complete" and a time of day at which to check the threshold. When that time of day is reached, the system checks the threshold percentage against the actual percentage to see if the production threshold is met. If the actual production percentage is greater than or equal to the threshold percentage, the threshold is met. Otherwise, the production threshold is violated and an alert pop-up is displayed.

Example:

For a run tag of "job 17A", the machines should be 50% complete by 11:30 AM. At 11:30 AM the system checks to see if the machines are 50% to the goal. If the actual percentage is less than 50%, an alert pop-up is displayed.

In addition, a pop-up window is presented to the user for viewing occurrences of production threshold violations. Preferably, a pop-up window automatically appears on the user's display when a threshold is crossed, in order to alert the user to the particular condition. Information that is preferably displayed in such a window includes: identification of insertion machine having an error, error type, number of occurrences of error, total errors, total time lost, and average reset time. Preferably, the item that caused the threshold to trigger the pop-up window will be highlighted. For example, if a threshold was set for 50% job completion by 12:00 noon and only 45% was actually achieved at that time then a production threshold violation pop-up window would appear on the user's display showing the current percentage complete thereby giving the operator an indication of job performance.

It is understood that additional production thresholds may be utilized without departing from the spirit and intent of the present invention.

Run Tag Editor

The "Run Tag Editor" screen of FIG. **16** is accessed via the "System Management" screen of FIG. **12**. The Run Tag Editor is a utility that allows the user to correct any run tag assignment mistakes made by the inserting machine operator. Since data is labeled in the present invention by a run tag, it is important that the run tag be correct. Therefore, it is desirable for the user to be able to make corrections to run tags.

Within the Run Tag Editor screen **126**, the user may search all of the run tags **162** in the database. The list includes run tags that have completed at least one run in the database. The user is prohibited from editing run tags for inserter runs that are in progress. Once the user has received the results of the run tag search, he may select one, many, or all run tags to edit. Upon selection, he will create a new run tag for the selected items either by typing a new tag **164** in the box provided on the screen or by selecting a previously used run tag from the list **166** above the box. He can then modify the tags by clicking the Modify button **168** located near the bottom of the screen. A pop-up will be presented to the user to show that the database will be altered and the user

must confirm this. Then, after the database is altered, the search list will be updated. Any errors that occur during this process will be displayed on the screen.

Backup/Restore

The data backup and restore functionality is divided into three (3) separate screens and each is accessed via the "System Management" screen of FIG. **12**. Such screens include the Backup screen (FIG. **17**), the Restore screen **180** (FIG. **18**), and the Scheduled Backup screen (not shown).

The user may choose to backup (FIG. **17**) one, many, or all of the following data types: run tag information **171**, operator setup information **172**, or system configuration information **173**. The user selects the data to be saved, selects the media (hard disk **174**, tape **175** or floppy **176**) and then executes the backup. Any errors that occur will be displayed on the screen.

The user may choose the Restore screen **180** illustrated illustrated in FIG. **18** to restore any data type from a backup file. First, the user selects the restoration media (hard disk **174**, tape **175** or floppy **176**). The selected media is accessed in order to present an archive list **181** or source list **182** of restorable data to the user. The user can then select to restore one, many, or all of the data in the backup file. Any errors that occur will be displayed on the screen for the user.

The user will be able to choose to schedule a backup for future or repeated execution. The user will select the data types and the media just as he would for a regular backup. The user would then select the time and interval at which the backup will occur. After making these selections, the user will be able to schedule the backup. The backup would then execute at the appropriate time. The user will also be able to use this screen to remove a command for a scheduled backup. The list at the top of the screen will display all currently scheduled backup parameters and the user would select any of them for viewing or deletion. Any errors that would occur during the setting of backup parameters will be displayed to the user. Any errors that would occur during a scheduled backup will be written to a log file.

A user can also define the valid list of operators for a given mail processing machine at a remote location. FIG. **19** illustrates the "Operator Definition" screen **123** selectable from the System Management screen of FIG. **12**. The Operator Definition screen **123** allows a system level user to identify operators of particular mail processing machines and define the level of access and/or control the operator will have with respect to the mail processing machine. The user will enter a unique Operator ID **190**, password **192**, and optional description **194** in the dialog box **196** on the upper portion of the screen. Next, the user will choose and assign the authority levels for this operator using the dialog box **198** in the lower center portion of the screen. This information is then communicated from the central location over a computer network to the specific mail processing machine in question. This information can be stored such that it can not be edited from the remote mail processing machine.

Hardware and Software Requirements

The present invention can be implemented in a variety of communications environments including a Local Area Network (LAN) and Wide Area Network (WAN) environments. The present invention can be implemented in communications environments utilizing TCP/IP communications protocol, such as the Internet, and environments utilizing SNA protocol. Hardware for implementing the present invention is generally consistent with typical personal computing equipment, and does not generally require special environmental conditions other than a typical office environment. Preferably, the present invention is implemented on an International Business Machines (IBM®) or IBM®-compatible personal computer and software capable of supporting a thin wire Ethernet TCP/IP environment. Even more preferable is a server based on an Intel® 486 or Pentium® processor and having at least sixteen (16) megabytes of memory to perform all functions efficiently, and having data storage capacity of at least four (4) gigabytes. Also preferred is a printer suitable for text and color graphical report printing; automatic back-up capability for data and configuration files; and trackball or mouse support. The present invention may be implemented via other computing devices, including, but not limited to, mainframe computing systems and mini-computers. It is preferable to use a high resolution color display; however, a standard personal computer monitor may be used.

The present invention may be written in various computer languages including, but not limited to, C++, Smalltalk, Java, and other conventional programming languages such as BASIC, FORTRAN and COBOL.

The present invention runs on current standard desktop computer platforms such as, but not limited to, Windows®, Windows 95®, Windows NT®, UNIX®, and OS/2®. The present invention utilizes, in part, many standard features of current desktop configurations, such as the ability to store data locally, connect to the Intemet, and display visual information.

Computer readable program code means is provided for receiving processing system operation related information from each of a plurality of mail insertion devices, and for representing each of the mail insertion devices as an interactive icon on a display connected to a data processing system. Each interactive icon has indicia associated therewith which displays the operation related information for a respective mail insertion device and changes appearance in response to a change in the operation related information. Computer readable program code means is provided for displaying selective operation related information about a respective mail insertion device in response to user actions, and for displaying operation related information for each mail insertion device in real time. Computer readable program code means is provided for selecting an interactive icon via an input device operatively connected to the data processing system, and for displaying operating status for the mail insertion device represented by the selected icon.

Computer readable program code means is provided for displaying the indicia surrounding an icon as a color selected from a plurality of colors, wherein each color represents an operational condition of a mail insertion device. Computer readable program code means is provided for displaying interactive icons in a manner depicting geographical locations of respective mail insertion devices. Computer readable program code means is also provided for adding, deleting, and modifying the location and appearance of the interactive icons.

The present invention is not limited in scope to systems, methods and computer program products for monitoring and controlling mail insertion machines. The present invention may be utilized for monitoring and controlling various mail processing systems including mail sorters, printers and other equipment related thereto. The foregoing is illustrative of the present invention and is not to be construed as limiting thereof. Although a few exemplary embodiments of this invention have been described, those skilled in the art will readily appreciate that many modifications are possible in the exemplary embodiments without materially departing from the novel teachings and advantages of this invention. Accordingly, all such modifications are intended to be included within the scope of this invention as defined in the claims.

In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures. Therefore, it is to be understood that the foregoing is illustrative of the present invention and is not to be construed as limited to the specific embodiments disclosed, and that modifications to the disclosed embodiments, as well as other embodiments, are intended to be included within the scope of the appended claims. The invention is defined by the following claims, with equivalents of the claims to be included therein.

That which is claimed:

1. A system for remote real-time monitoring of at least one mail inserter machine comprising:

(a) receiving means located remotely from said mail inserter machine for receiving operation related information from said mail inserter machine;

(b) monitoring means operatively connected to said receiving means for monitoring the operation related information received from said mail inserter machine and updating display of the operation related information in real time; and

(c) transmission means for sending data to said mail inserter machine.

2. The system of claim 1 wherein said receiving means and monitoring means are centrally located not necessarily proximate to said mail inserter machine.

3. The system of claim 1 wherein said receiving and monitoring means are coupled to a networked communications system for exchanging data with said mail inserter machine.

4. The system of claim 1 wherein said monitoring means updates display of the operation related information on a computer display screen at a rate of at least about one update per thirty seconds.

5. The system of claim 1 wherein said receiving means receives operation related information from a plurality of mail inserter machines located at a plurality of geographically separated sites.

6. The system of claim 1 wherein the operation related information includes mail inserter machine organizational information, site information, machine information, utility information, and system administration information.

7. The system of claim 6 wherein the mail inserter machine organizational information includes mail inserter machine site addition information, mail inserter machine site deletion information, and mail inserter machine site edit information, and mail inserter machine site job goal information.

8. The system of claim 6 wherein the mail inserter machine site information includes production information, goal information, job information, fault information, and alarm information with respect to a specified mail inserter machine site.

9. The system of claim 6 wherein the mail inserter machine machine information includes production information, goal information, job information, fault information, and alarm information with respect to said mail inserter machine.

10. The system of claim 6 wherein the mail inserter machine utility information includes alarm information, user administration information, shift information, language information, threshold setting information, and run-time manager information with respect to said mail inserter machine.

11. The system of claim 6 wherein the mail inserter machine system administration information includes report information, machine setup information, backup and restore information, and run tag editor information with respect to said mail inserter machine.

12. A method for real-time monitoring of at least one mail inserter machine comprising:

(a) receiving operation related information from a mail inserter machine from a location remote from said mail inserter machine;

(b) monitoring the operation related information received from said mail inserter machine, and updating display of the operation related information in real time; and

(c) transmitting data to said mail inserter machine.

13. The method of claim 12 wherein said receiving means and monitoring means are centrally located not necessarily proximate to any of said mail inserter machine.

14. The method of claim 12 wherein said receiving and monitoring means are coupled to a networked communications system for exchanging data with said mail inserter machine.

15. The method of claim 12 wherein receiving the operation related information includes receiving the operation related information from a plurality of mail inserter machines located at a plurality of geographically separated sites.

16. The method of claim 12 wherein updating display of the operation related information in real time includes updating display of the operation related information at a rate of at least about one update per thirty seconds.

17. The method of claim 12 wherein the operation related information includes mail inserter machine organizational information, site information, machine information, utility information, and system administration information.

18. The method of claim 17 wherein the mail inserter machine machine information includes production information, goal information, job information, fault information, and alarm information with respect to said mail inserter machine.

19. The method of claim 17 wherein the mail inserter machine utility information includes alarm information, user administration information, shift information, language information, threshold setting information, and run-time manager information with respect to said mail inserter machine.

20. The method of claim 17 wherein the mail inserter machine system administration information includes report information, machine setup information, backup and restore information, and run tag editor information with respect to said mail inserter machine.

21. The method of claim 17 wherein the mail inserter machine organizational information includes mail inserter machine site addition information, mail inserter machine site deletion information, mail inserter machine device site edit information, and mail inserter machine site job goal information.

22. The method of claim 17 wherein the mail inserter machine site information includes production information, goal information, job information, fault information, and alarm information with respect to a specified mail inserter machine site.

23. A remote real-time monitoring device for remotely monitoring at least one mail inserter machine, said remote real-time monitoring device comprising:

(a) operating parameter monitoring means located remotely from said at least one mail inserter machine for remotely monitoring the operating parameters of said at least one mail inserter machine and for updating display of the operating parameters in real time; and

(b) computer network communications means coupled to the operating parameter monitoring means for transferring data between the remote real-time monitoring device and said at least one mail inserter machine.

24. The device of claim **23** further comprising operator setup means for remotely creating, maintaining, and communicating a list of valid mail inserter machine operators for said at least one mail inserter machine.

25. The device of claim **23** further comprising run-tag editing means for remotely accessing, editing, and re-saving stored run-tag data for a specified job on said at least one mail inserter machine.

26. The device of claim **25** wherein said run-tag editing means is capable of searching and selecting individual and multiple run-tags from a stored run-tag database.

27. The device of claim **23** further comprising alarm threshold setting means for setting alarm threshold levels of tolerable error counts for specified types of errors for a specified job on said at least one mail inserter machine.

28. The device of claim **27** wherein said specified types of errors include aim errors, insert feeder errors, envelope feeder errors, finishing errors, and delivery errors.

29. The device of claim **27** further comprising production threshold setting means for setting production threshold levels of output goals to be completed within or by a specified time for a specified job on said at least one mail inserter machine.

30. The device of claim **29** wherein said output goals represent the number of envelopes processed at a specified time.

31. The device of claim **29** wherein said operating parameter monitoring means monitors the production threshold levels and alarm threshold levels of said at least one mail inserter machine in real-time and outputs a message when said threshold levels have been violated.

32. A computer program product comprising computer-executable instructions embodied in a computer-readable medium for performing steps comprising:

(a) receiving operation related information from at least one mail inserter machine from a location remote from said at least one mail inserter machine; and

(b) updating display of the operation related information received from said mail inserter machine in real time.

33. The computer program product of claim **32** wherein receiving the operation related information includes receiving the operation related information over a TCP/IP network.

34. The computer program product of claim **32** comprising sending data to said mail inserter machine.

35. The computer program product of claim **32** wherein receiving the operation related information includes receiving the operation related information from a plurality of mail inserter machines located at geographically separated sites.

36. The computer program product of claim **32** wherein updating display of the operation related information in real time includes updating display of the operation related information at a rate of at least about one update per thirty seconds.

37. A system for remote real-time monitoring of at least one mail sorting machine, said system comprising:

(a) receiving means for receiving operation related information from said mail sorting machine from a location remote from said at least one mail sorting machine; and

(b) monitoring means operatively connected to said receiving means for monitoring the operation related information received from said mail sorting machine and updating display of the operation related information in real time.

38. The system of claim **37** comprising transmission means for sending data to said mail sorting machine.

39. The system of claim **38** wherein said receiving means receives operation related information from a plurality of mail sorting machines located at a plurality of sites geographically separated from each other.

40. The system of claim **37** wherein said monitoring means updates display of the operation related information on a computer display screen at a rate of at least about one update per thirty seconds.

41. A method for remote real-time monitoring of at least one mail sorting machine comprising:

(a) receiving operation related information from a mail sorting machine from a location remote from said mail sorting machine; and

(b) updating display of the operation related information in real time.

42. The method of claim **41** comprising transmitting data to said mail sorting machine.

43. The method of claim **41** wherein receiving the operation related information includes receiving the operation related information from a plurality of mail sorting machines located at geographically separate sites.

44. The method of claim **41** wherein updating display of the operation related information in real time includes updating display of the operation related information at a rate of at least about one update per thirty seconds.

45. A computer program product comprising computer-executable instructions embodied in a computer-readable medium for performing steps comprising:

(a) receiving, from a user, a threshold percentage value relating to percentage completion of a job by at least one mail processing machine and a time value for checking whether the mail processing machine has reached the percentage;

(b) receiving at least one measured percentage value indicating the percentage completion of a job by the mail processing machine from a location remote from the mail processing machine;

(c) when the time value is reached, determining whether the measured percentage value equals the threshold percentage value; and

(d) if the measured percentage value is less than the threshold percentage value, displaying an alarm to the user.

46. The computer program product of claim **45** wherein receiving at least one measured percentage value includes receiving at least one measured percentage value from one or more mail inserter machines.

47. The computer program product of claim **45** wherein receiving at least one measured percentage value includes receiving at least one measured percentage value from one or more mail sorting machines.

48. The computer program product of claim **45** wherein alerting the user includes displaying a pop-up window to the user.

49. The computer program product of claim **48** wherein the pop-up window displays the measured percentage value.

50. The computer program product of claim **45** comprising updating display of the measured percentage value in real time.

51. The computer program product of claim **50** wherein updating display of the measured percentage value includes updating display at a rate of at least about one update per thirty seconds.

52. A run tag editor comprising computer-executable instructions embodied in a computer-readable medium for performing steps comprising:

(a) searching a database for run tags indicative of jobs performed by at least one mail processing machine;

(b) displaying run tags extracted from the database to a user;

(c) receiving input from a user allowing modification of the displayed run tags and addition of new run tags; and

(d) storing the modified or new run tags in the database.

53. The computer program product of claim **52** wherein run tags are indicative of jobs performed by a mail inserter machine.

54. The computer program product of claim **52** wherein the run tags are indicative of jobs performed by a mail sorting machine.

* * * * *